

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Accesibilidad relacionada con displays 7-segmentos y paneles indicadores mediante Android

Máster en INGENIERÍA INFORMÁTICA

Autor: BAYÓN BENEGAS, Marc

Tutor: BOEMO SCALVINONI, Eduardo

FECHA: junio, 2017

Índice de contenidos

Índice de contenidos.....	5
Índice de ilustraciones	9
Índice de tablas.....	11
1 Capítulo 1: Introducción	15
1.1 Ideas de partida.....	17
1.1.1 Accesibilidad como problema presente	17
1.1.2 Estudio de la ceguera como barrera de entorno.	18
1.1.3 Domótica como punto de partida	19
1.1.4 ¿Qué servicios nos ofrece la domótica?.....	19
1.1.5 Movilidad y OCR como herramientas base.	21
1.1.6 Nuestra propuesta	22
1.2 ¿De qué trata nuestro proyecto?	23
1.2.1 Objetivo general	23
1.2.2 Objetivos específicos.....	23
1.3 Estudio Histórico	24
1.4 Sistemas de vitrocerámicas para invidentes en la actualidad.	26
2 Capítulo 2: Planificación	31
2.1 Planificación de costes, el modelo COCOMO	33
2.2 Aplicación del modelo	34
2.3 Cálculos y resultados del modelo	36
2.4 Planificación temporal	37
3 Capítulo 3: Elicitación de requisitos	41
3.1 Organizaciones y participantes	43
3.2 Objetivos del Proyecto	45
3.3 Requisitos	50
3.3.1 Requisitos de información.....	50
3.4 Requisitos funcionales.....	60
3.4.1 Actores.....	60

3.4.2	Diagramas de casos de uso.....	61
3.4.2.1	Representación del sistema.....	61
3.4.2.2	Subsistema de acciones del usuario.....	62
3.4.2.3	Subsistema de acciones del superusuario.....	63
3.4.3	Descripción de los casos de uso.....	64
3.5	Requisitos no funcionales.....	86
3.6	Reglas de negocio.....	88
3.7	Matrices de Rastreabilidad	92
4	Capítulo 4: Arquitectura del sistema. Detalles de diseño.....	97
4.1	Introducción.....	99
4.2	Visión General.....	101
4.2.1	Subsistema 1: Captura, tratamiento y procesamiento de imágenes. 102	
4.2.2	Subsistema 2: Comunicación con unidad de control inteligente, gestión de órdenes y manejo de actuadores.	106
4.3	Definición de Tesseract® como motor OCR y fase de entrenamiento.	107
4.3.1	Descripción de la fase de entrenamiento.....	107
4.3.1.1	Preparación del sistema operativo.....	107
4.3.1.1.1	Instalar Leptonica	108
4.3.1.1.2	Instalar Tesseract.....	108
4.3.1.2	Descripción del procedimiento de entrenamiento.....	109
4.4	Arquitectura de la aplicación móvil.....	113
4.4.1	Back-end de la aplicación.....	113
4.4.1.1	OpenCV.....	114
4.4.1.2	"Tess-two" Wrapper Tesseract para Android.	114
4.4.2	Front-End de la aplicación	115
4.5	Arquitectura de la unidad de control inteligente	118
4.5.1	Protocolo de comunicación, capa física y de transporte: Bluetooth, L2CAP	119
4.5.2	Protocolo de comunicación, capa de aplicación: Binary Protocol.	120
4.5.2.1	Lista de comandos implementados.	121

5	Capítulo 5: Justificación de tecnologías usadas y preparación del entorno de desarrollo	125
5.1	Introducción.....	127
5.2	Razones por las que se ha optado por Android como sistema operativo para el desarrollo.	129
5.3	Razones por las que se optado por Rapsberry Pi como microcontrador.....	130
5.3.1	Elección de Raspbian como sistema operativo de Raspberry Pi.....	131
5.4	Elección de Tesseract como motor OCR.....	132
5.5	Razones por las que se usado OpenCV como capa de tratamiento de imágenes.	133
5.6	Razones por la que se ha usado Python en Raspberry Pi para el servicio bluetooth.	134
5.7	Uso de Java para la interfaz de usuario de la aplicación Android.	135
5.8	Eclipse como IDE para Android	136
5.8.1	Configuración de eclipse para trabajar con Android, Tesseract y OpenCV.	137
5.8.1.1	Compilar Tess-Two.....	138
6	Capítulo 6: Implementación.....	143
6.1	Introducción.....	145
6.2	Instalación de la aplicación en un dispositivo Android.....	146
6.2.1	Modo sencillo a través de Google Play.	146
6.2.2	Modo desarrollador a través de Eclipse.....	147
6.3	Vistazo a la implementación de la aplicación Android.....	148
6.3.1	Estructura de clases de la aplicación principal.	148
6.3.2	Splash activity	148
6.3.3	Main Activity	150
6.3.3.1	Implementación del cliente bluetooth.	150
6.3.3.2	Implementación de los mensajes de comunicación entre aplicación móvil y dispositivo bluetooth.	153
6.3.3.3	Vistazo a la interfaz final de Main Activity.	155

6.3.4	Gesture Activity	156
6.3.5	OCR Activity.....	158
6.3.5.1	Inicialización de la actividad	158
6.3.5.2	Análisis de cada fotograma capturado	160
6.3.5.3	Detección de cuadrados por aproximación de polinomios y detección de ángulo. 163	
6.3.5.4	Implementación de la corrección de ángulos "Deskew"	165
6.4	Implementación del servidor bluetooth en Raspberry Pi.....	167
6.4.1	Instalación de Raspbian usando NOOBS.....	167
6.4.2	Preparación del entorno.	169
6.4.3	Vistazo al código fuente.....	170
6.4.4	Mapa de GPIO Ports usados y esquemático del circuito.....	172
7	Capítulo 7: Pruebas del sistema	175
7.1	Introducción.....	177
7.2	Hardware utilizado para el plan de pruebas.	177
7.3	Plan de pruebas a realizar	179
7.3.1	Pruebas de lectura OCR	179
7.3.2	Test de conexión Bluetooth	184
7.3.3	Test sobre el modo accesibilidad con Android Gesture.....	184
7.3.4	Prueba de esfuerzo en Raspberry Pi	185
Anexo1:	Manual de Usuario	189
A.1	Introducción.....	191
A.2	Requisitos del teléfono inteligente	191
A.3	Instalación y puesta en marcha.....	192
A.4	Modo de empleo y funcionamiento	192
A.5	Modo con botones	193
A.6	Modo accesibilidad (Gesture)	194
A.7	Modo OCR	195
Apéndice 1:	Referencias	197

Índice de ilustraciones

Ilustración 1: Accesibilidad Universal, reproducido de [29]	17
Ilustración 2: Ceguera, reproducido parcialmente de: [30]	18
Ilustración 3: Proceso OCR de conversión de una fotografía de una matrícula a caracteres, reproducido de [1]	21
Ilustración 4: Prototipos de vitrocerámicas para este proyecto	22
Ilustración 5: Uno de los primeros prototipos de cocina eléctrica, reproducido de [34]	24
Ilustración 6: Cocina de inducción, reproducido de [31]	25
Ilustración 7: Cooking without Looking, reproducido de [32]	26
Ilustración 8: Touch&Turn, reproducido de [33]	27
Ilustración 9: Sentino Cook System, reproducido de [2]	28
Ilustración 10: Diagrama del sistema y subsistemas principales	61
Ilustración 11: Subsistema de acciones del usuario	62
Ilustración 12: Subsistema de acciones del superusuario	63
Ilustración 13: Arquitectura del sistema	99
Ilustración 14: División de subsistemas.....	101
Ilustración 15: Detección automática de la región de interés por detección de cuadrados.....	102
Ilustración 16: Recorte y Binarización método OTSU.....	103
Ilustración 17: Suavizado de la imagen a partir de una imagen binaria.	103
Ilustración 18: Ejemplo de deskew de una imagen realizada por la aplicación móvil.....	104
Ilustración 19: Eliminación de Líneas.	105
Ilustración 20: Procesamiento OCR	105
Ilustración 21: Protocolo de comunicación bluetooth para este TFM.....	106
Ilustración 22: Muestras recogidas por la cámara del teléfono con calidad media.	109
Ilustración 23: Fichero de entrenamiento de tesseract.....	110
Ilustración 24: Visión del programa generador de cajas vietOCR	111
Ilustración 25: Backend de la aplicación	113
Ilustración 26: Esquema de navegabilidad de la aplicación.	116
Ilustración 27: Wireframe final de la aplicación móvil.	117
Ilustración 28: Raspberry Pi 3 Modelo B.....	118
Ilustración 29: Evolución de la cuota de mercado de los sistemas operativos para teléfonos inteligentes, reproducido de [19]	129
Ilustración 30: Captura de pantalla del sistema operativo Raspbian.	131

Ilustración 31: Logotipo y usos de Tesseract sobre Google Translator, reproducido de [35].....	132
Ilustración 32: Evolución del uso de los lenguajes de programación más populares, reproducido de [24]	135
Ilustración 33: Eclipse	136
Ilustración 34: Asistente de configuración del SDK.	137
Ilustración 35: Configuración de NDK de Android en Eclipse.	138
Ilustración 36: Configuración de compilación del código de Tess Two usando NDK.	139
Ilustración 37: Logotipo de la aplicación	145
Ilustración 38: Go Kitchen en Google Play.....	146
Ilustración 39: OpenCv en Google Play	146
Ilustración 40: Esquema de navegación de la aplicación móvil	148
Ilustración 41: Splash Screen.....	149
Ilustración 42: Pantalla Main Activity.....	155
Ilustración 43: Aplicación Gesture builder.....	156
Ilustración 44: Instalación de Raspbian	168
Ilustración 45: Entorno de escritorio Raspbian con Pixel	168
Ilustración 46: Esquemático del circuito de prueba.	172
Ilustración 47: Resultados del Test de luminosidad.....	179
Ilustración 48: Lanzador de la aplicación en Android.....	192
Ilustración 49: Detalle de la pantalla principal de la aplicación.....	193
Ilustración 50: Detalle del modo gesture.....	194
Ilustración 51: Detalle del modo OCR.....	195

Índice de tablas

Tabla 1: Modelo COCOMO, Modelo básico	35
Tabla 2: Estimación inicial del proyecto	38
Tabla 3: Tiempo total empleado en el proyecto	38
Tabla 4: Error Relativo cometido en la estimación inicial	38
Tabla 5: Descripción Organización 1	43
Tabla 6: Descripción Organización 2	43
Tabla 7: Descripción Organización 3	43
Tabla 8: Participante 1 - Marc Bayón.....	44
Tabla 9: Participante 2 - Sergio Geninatti.....	44
Tabla 10: Participante 3 - Eduardo Boemo	44
Tabla 11: OBJ-0001 - Gestión de la seguridad	45
Tabla 12: OBJ-0002 Gestión de la Tele-lectura	46
Tabla 13: OBJ-0003 - Gestión de la comunicación y tratamiento de datos.....	47
Tabla 14: OBJ-0004 - Gestión de la Interactuación Usuario-Máquina	48
Tabla 15: OBJ-0015 - Gestión del ahorro energético	49
Tabla 16: IRQ-0001 Vitrocerámica	50
Tabla 17: IRQ-0002 Gestos.....	51
Tabla 18: IRQ-0003 Sensores	52
Tabla 19: IRQ-0004 Actuadores.....	53
Tabla 20: IRQ-0005 Patrones numéricos	54
Tabla 21: IRQ-0006 Estadísticas y Autoaprendizaje	55
Tabla 22: IRQ-0007 Alertas.....	56
Tabla 23: IRQ-0008 Estado.....	57
Tabla 24: IRQ-0009 Mensajes de Voz Sintética	58
Tabla 25: IRQ-0010 Modos de Ahorro Energético	59
Tabla 26: Actor 1 Usuario.....	60
Tabla 27: Actor 2 Superusuario.....	60
Tabla 28: UC-0001 Leer un dígito del display 7 segmentos de la vitrocerámica.....	64
Tabla 29: UC-0002 Encender Vitrocerámica.....	65
Tabla 30: UC-0003 Apagar Vitrocerámica	66
Tabla 31: UC-0004 Disminuir potencia.....	67
Tabla 32: UC-0005 Aumentar potencia	68
Tabla 33: UC-0007 Añadir tiempo de cocción	69
Tabla 34: UC-0020 Eliminar tiempo de cocción	70
Tabla 35: UC-0008 Orientar Dispositivo Móvil	71
Tabla 36: UC-0009 Emparejar vitrocerámica y aplicación móvil.....	72
Tabla 37: UC-0014 Desemparejar vitrocerámica y aplicación móvil.....	73

Tabla 38: UC-0010 Calibrar cámara de lectura	74
Tabla 39: UC-0011 Añadir plantillas al motor OCR	75
Tabla 40: UC-0015 Eliminar plantillas del motor OCR	76
Tabla 41: UC-0012 Calibrar el software de mejora de imagen.....	77
Tabla 42: UC-0013 Obtener Información del dispositivo	78
Tabla 43: UC-0016 Lanzar una alarma	79
Tabla 44: UC-0017 Añadir una configuración de ahorro energético	80
Tabla 45: UC-0018 Eliminar una configuración de ahorro energético.....	81
Tabla 46: UC-0019 Modificar una configuración de ahorro energético	82
Tabla 47: UC-0021 Obtener informes.....	83
Tabla 48: UC-0022 Desactivación por llave remota o mecánica	84
Tabla 49: UC-0023 Activación de la llave remota o mecánica.....	85
Tabla 50: NFR-0001 Protección de datos.....	86
Tabla 51: NFR-0002 Reconexión	86
Tabla 52: NFR-0003 Autenticación.....	87
Tabla 53: CRQ-0001 Distancia máxima entre dispositivos.....	88
Tabla 54: CRQ-0002 Límite de conexiones vinculadas.....	89
Tabla 55: CRQ-0003 Derechos de superusuario para configuración del sistema ...	90
Tabla 56: CRQ-0004 Tiempo máximo para reintentar la pérdida de una conexión	90
Tabla 57: CRQ-0005 Tiempo máximo de funcionamiento a pleno rendimiento.....	91
Tabla 58: TRM-0001 Matriz relación objetivos del proyecto con Requisitos de información.....	92
Tabla 59: TRM-0002 Matriz relación entre casos de uso y requisitos de información	93
Tabla 60: TRM-0003 Matriz relación entre casos de uso y objetivos	94
Tabla 61: Binary Protocol.....	120
Tabla 62: Lista de Comandos disponibles.	121
Tabla 63: Software y Hardware necesario para realizar el desarrollo.....	127
Tabla 64: Relación de señales y GPIO Ports.....	172
Tabla 65: Resultados del test de distancia de lectura.....	180
Tabla 66: Resultados del test de lectura en distintas posiciones.	181
Tabla 67: Resultado del test con distintas tipografías.....	182
Tabla 68: Resultado del test de detección de la ROI.....	183
Tabla 69: Resultado del test de conexión bluetooth.....	184
Tabla 70: Resultados del test de Android Gesture.	184
Tabla 71: Resultados del test de esfuerzo en Raspberry Pi.	185
Tabla 72: Requisitos de software y hardware.	191

CAPÍTULO 1: INTRODUCCIÓN

1.1.2 Estudio de la ceguera como barrera de entorno.

La **ceguera** como concepto es una diversidad de tipo sensorial que consiste en la pérdida total o parcial del sentido de la vista. Existen varios tipos de ceguera parcial dependiendo del grado y del tipo de pérdida de visión, como la visión reducida, el escotoma, la ceguera parcial (de un ojo) o el daltonismo:

Ciegos: son aquellos sujetos que tienen sólo percepción de luz, sin proyección, o aquellos que carecen totalmente de visión. Desde el punto de vista educativo, ciego es aquel que aprende sistema Braille y no puede utilizar su visión para adquirir ningún conocimiento, aunque la percepción de la luz pueda ayudarle para sus movimientos y orientación.

Ciegos parciales: son aquellos sujetos que mantienen unas posibilidades visuales mayores, tales como capacidad de percepción de la luz, percepción de bultos y contornos, algunos matices de color, etc.

Personas con baja visión: son los que mantienen un resto visual que les permite ver objetos a pocos centímetros. A estos no se les debe llamar nunca ciegos ni se les debe educar como tales, aunque tengan que aprender procedimientos "táctiles" para aumentar sus conocimientos.

Limitados visuales: son los que precisan, debido a sus dificultades para aprender, una iluminación o una presentación de objetos y materiales más adecuadas, utilizando lentes, aumentando la iluminación, etc.



Ilustración 2: Ceguera, reproducido parcialmente de: [30]

1.1.3 Domótica como punto de partida

El término **domótica** proviene de la unión de las palabras "domus" (que significa casa/hogar en latín) y "tica" (de automática, palabra en griego, 'que funciona por sí sola').

En Europa se usa la palabra de origen francés "domótica" para referirse a la automatización de la vivienda. En EEUU se utiliza el término "smart house", vivienda o casa inteligente. Aparte de estos vocablos, se emplean otras expresiones para referirnos a este campo, como son "home systems", sistemas domóticos, o "home automation", automatización del hogar.

Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación. Pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado.

1.1.4 ¿Qué servicios nos ofrece la domótica?

Para nuestro sistema, los principales objetivos que buscamos son los siguientes:

- **Seguridad**

La seguridad es la principal faceta a tener en cuenta. En función de los sensores/actuadores instalados podremos, entre otras funciones, realizar lo siguiente:

- Activar cualquier alarma ante el valor anómalo de un sensor/actuador
- Activar y desactivar a voluntad la toma de corriente para evitar accidentes fortuitos
- Autodiagnóstico
- Sistemas de seguridad para personas discapacitadas, reduciendo riesgos cotidianos con sistemas inteligentes.

- **Confort y usabilidad**

Con un sistema domótico, se puede reducir la carga de trabajo que un usuario debe hacer, aumentando el confort y reduciendo los riesgos que entraña las actividades cotidianas que una persona discapacitada pueda realizar:

- Control y programación de electrodomésticos.
- Garantizar accesibilidad universal en interfaces de usuario para sistemas cotidianos de uso.
- Hacer la tecnología controlable y portable, diseñarla para dispositivos móviles.
- Convertir un sistema complejo cotidiano en algo sencillo.
- Agrupar subsistemas en uno solo que permita el control total de manera que aumente el confort y la usabilidad.
- Fácil acceso a telemetría y datos.

- **Ahorro de energía**

Un sistema domótico debe tener como tercera premisa el ahorro de energía. En este aspecto, primeramente, se deberá analizar las fases de funcionamiento del sistema, así como el rendimiento necesario mínimo para realizar correctamente las funciones cotidianas. Es por ello que se requiere de lógica que sea capaz de analizar el entorno donde se encuentra y comportarse de tal manera que el rendimiento y ahorro sean máximos en la mayor parte del funcionamiento del sistema.

- **Ventajas en telemetría y comunicaciones.**

Un buen control de los sistemas automáticos requiere disponer de un buen sistema de comunicaciones estable que garantice el manejo y la interconexión de diferentes dispositivos. Teléfonos móviles, ordenadores, tabletas, mandos a distancia... son solo alguno de los dispositivos que nos permiten, a través de la comunicación, que estos sistemas sean versátiles y fácilmente ampliables.

1.1.5 Movilidad y OCR como herramientas base.

Uno de los pilares fundamentales hoy en día en cualquier sistema inteligente es la movilidad. Tener la tecnología en la palma de tu mano hace accesible y ágil el control de cualquier elemento a distancia.

Cualquier dispositivo móvil viene incorporado con una cámara que permite tomar luz del exterior y transformar mediante un sensor, dicha luz en píxeles.

Partiendo de esa idea, es posible usar dicha imagen como elemento de lectura, haciendo uso de la propia cámara del dispositivo móvil y de algoritmos que permitan reconocer ciertos patrones de píxeles y extraer información en forma de caracteres alfanuméricos. Es lo que se conoce como OCR.

OCR son las siglas de "Optical Character Recognition" y define el proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de una imagen símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en forma de datos. Así podremos interactuar con estos mediante un programa de edición de texto o similar.

Combinado un sistema domótico, la tecnología OCR ofrece un complemento extra y un valor añadido a cualquier dispositivo en lo referente a la accesibilidad proporcionando un elemento importante de lectura e interpretación del entorno que puede ser utilizado para múltiples usos. Uno de ellos es el referente al descrito en el siguiente punto de nuestra propuesta.



Ilustración 3: Proceso OCR de conversión de una fotografía de una matrícula a caracteres, reproducido de [1]

1.1.6 Nuestra propuesta

En nuestra propuesta nos centraremos en dar una solución domótica diseñada para personas con discapacidad visual o con baja visión. Se ha detectado que en las placas vitrocerámicas actuales existentes en el mercado un problema de accesibilidad para este tipo de usuarios.

El problema reside en que la propia placa vitrocerámica no dispone de ningún tipo de rugosidad para su manejo de tal manera que no es factible usar dicho electrodoméstico sin tener la posibilidad de ver. Dada la singularidad de la vitrocerámica, y debido a su alta temperatura (alcanza fácilmente los 500°C), se desea diseñar una solución que elimine esta barrera de manera sencilla. También cabe destacar que la vitrocerámica no posee ninguna señal acústica significativa. Tan solo en algunos modelos, existe un pitido cada vez que se aumenta o disminuye la temperatura, sin llegar a tener pleno significado para una persona con discapacidad visual o baja visión. Otro atributo visual que poseen esta placa para indicar la potencia de funcionamiento, por regla general es un simple display 7 segmentos que indica un número del 0 al 9.

Este trabajo fin de máster pretende aportar soluciones para el diseño de un lector óptico que descodifique los indicadores de la cocina y los transforme en una lectura de voz sintética en un teléfono Android. También se propone complementariamente del uso de comandos a partir de gestos para poder manejar dicho electrodoméstico a través del propio teléfono.



Ilustración 4: Prototipos de vitrocerámicas para este proyecto

1.2 ¿De qué trata nuestro proyecto?

1.2.1 Objetivo general

El **objetivo general** del proyecto es realizar un sistema de monitorización y control de una placa vitrocerámica para eliminar barreras en el colectivo de personas invidentes, y que además sea capaz de identificar mediante lector óptico los indicadores de la propia vitrocerámica y los transforme en lectura de voz sintética.

1.2.2 Objetivos específicos

El proyecto consistirá con dos partes diferenciadas:

Parte 1: Aplicación Android. Deberá existir una aplicación desarrollada para sistemas Android que deberá realizar las siguientes funciones:

- Lectura a través de dispositivo óptico (cámara del dispositivo) de un display 7 segmentos.
- Reproducción de la lectura a través de una voz sintética
- Uso de Bluetooth como sistema de comunicación entre la placa vitrocerámica y el teléfono.
- Uso de comandos "gesture" para mandar órdenes a la placa vitrocerámica.
- Uso de comandos de voz para mandar órdenes a la placa vitrocerámica.

Parte 2: Vitrocerámica Inteligente. Este electrodoméstico debe disponer de las siguientes funciones:

- Diseño e integración de actuadores inteligentes en objetos domésticos.
- Estados de reposo seguro de los actuadores.
- Opciones de desactivación.
- Llave maestra de desactivación remota/local y mecánica/electrónica.
- Técnicas de auto-test.
- Minimización de consumo de potencia.
- Aislaciones galvánicas.

1.3 Estudio Histórico

Si echamos la vista atrás, tenemos que partir del concepto de **cocina eléctrica**. Su base está ligada al invento de las estufas eléctricas. En 1892, una década después de que Edison diera a conocer la lámpara incandescente, los británicos **R. E. Crompton** y **J. H. Dowsing** patentaron la primera estufa eléctrica para uso doméstico. El nuevo aparato consistía en un alambre de alta resistencia enrollado varias veces alrededor de una placa rectangular de hierro. El alambre, que al conducir la electricidad adquiría un brillo blanco anaranjado, estaba situado en el centro de una pantalla parabólica que concentraba y difundía el calor en un haz.



Ilustración 5: Uno de los primeros prototipos de cocina eléctrica, reproducido de [34]

No tardaron en aparecer modelos perfeccionados de estufas eléctricas, y dos de los más notables fueron el de 1906, debido al inventor **Albert Marsh**, de Illinois (EE. UU.), cuyo elemento radiante, de níquel y cromo, podía alcanzar temperaturas al rojo blanco sin fundirse; y la estufa británica de 1912, que sustituyó la pesada placa de hierro en la que se enrollaba el alambre calefactor por un elemento ligero de arcilla refractaria, con lo que se consiguió la primera estufa eléctrica portátil realmente eficaz.

Tras el invento de las estufas eléctricas, que usaban la electricidad por efecto Joule para el calentamiento, se aplicó la técnica a los fogones, mediante resistencias arrolladas helicoidalmente en una base de material cerámico y, más adelante, mediante resistencias blindadas o embutidas en una placa metálica.

A partir de la perfección de estos modelos, se consiguió el modelo de cocina vitrocerámica que conocemos hoy en día, en los cuales la resistencia está embutida en un vidrio especial. Se les denomina comúnmente cocina vitrocerámica.

Las hay de varios tipos, vitrocerámica normal, halógena o de inducción. Ésta última no calienta la superficie del vidrio, solo produce calor a objetos metálicos colocados encima.



Ilustración 6: Cocina de inducción, reproducido de [31]

Como se observa en este breve estudio histórico, no existe ninguna cocina adaptada para personas con discapacidad visual.

Nuestra idea, como ya se ha descrito anteriormente, consiste en proporcionar inteligencia a este tipo de cocinas, creando el prototipo de cocina inteligente que debería ser el próximo paso en el escalafón de este breve estudio histórico. Nuestro principal paso adelante va encaminado también a hacer estos electrodomésticos más accesibles a personas con movilidad reducida.

1.4 Sistemas de vitrocerámicas para invidentes en la actualidad.

En este capítulo haremos un breve repaso a los sistemas actuales que están en el mercado para personas invidentes, y también de aquellos que están en vías de desarrollo:

1) Cooking without looking.

Este sistema, diseñado por [Nick Imsand](#), no es propiamente un sistema inteligente de vitrocerámicas adaptada para personas con discapacidad visual. En este caso se trata de un adaptador que se coloca sobre la vitrocerámica facilitando la localización de los fogones.



Ilustración 7: Cooking without Looking, reproducido de [32]

Las ventajas de este sistema es que simple y sencillo. No representa ningún tipo de dificultad su instalación, y es fácil de usar.

La principal desventaja es que no es un sistema inteligente. Es por ello que presenta deficiencias en la seguridad. Solo ayuda en la guía de

colocación de los utensilios de cocción. La persona incidente sigue sin tener información del estado de la vitrocerámica. Tampoco este sistema muestra ningún tipo de señal extra al usuario invidente que le percate de algún mal uso o alarma.

2) Touch&Turn

Desarrollada en cooperación con "RNIB" y "Action for Blind People", Touch&Turn es un sistema de vitrocerámica que ofrece una interfaz adaptada para personas con visibilidad reducida. El punto fuerte es su interfaz de uso, realmente práctica y que solventa muchos de los problemas de seguridad tales como el contacto con una superficie caliente.



Ilustración 8: Touch&Turn, reproducido de [33]

La principal ventaja es la adaptación de los mandos y la protección de la zona caliente de la vitrocerámica para prevenir accidentes.

La desventaja es que no es inteligente, sigue siendo un sistema tradicional que no responde pos si solo ante alarmas, proximidad, y aumento de confort.

3) Sentino - Induction Cook Top for the Blind:

Sentino [2] es una placa de inducción totalmente pensada para personas con discapacidad visual. Sentino usa diseños adaptados para facilitar la tarea a usuarios invidentes.

Sentino usa controles lineales, con superficies rugosas o elevadas, en la zona de los mandos y sobre las superficies calientes. Posee un sistema de sonidos artificiales para guiar a la persona y además usa un sistema de inducción que sólo calienta el objeto donde se encuentra el alimento. Este electrodoméstico es también respetuoso con el medio ambiente garantizando un bajo consumo en su uso.

La principal ventaja es básicamente su interfaz de usuario. Es universal, y apta para personas con discapacidad visual. Emula también un sonido artificial una cocina de gas, para que la persona pueda ser capaz de saber cuándo está encendida la cocina. El sonido que emite ofrece también información acerca de la temperatura a la que se encuentra la comida que se está preparando. Al ser también una cocina de inducción, se evita también que la persona

Por otro lado, no ofrece una interfaz telemática de uso. Su control debe hacerse in situ. Tampoco posee ningún elemento de seguridad que permita desconectar la placa en caso de un uso no adecuado.



Ilustración 9: Sentino Cook System, reproducido de [2]

CAPÍTULO 2: PLANIFICACIÓN

2.1 Planificación de costes, el modelo COCOMO

El **Modelo Constructivo de Costes** [3] (o **COCOMO**, por su acrónimo del inglés Constructive Cost Model), es un modelo de estimación de costes de software que incluye tres sub-modelos, donde cada uno ofrece un nivel de detalle y aproximación cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado. Fue desarrollado por Barry W. Boehm a finales de los 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981).

Pertenece a la categoría de modelos basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto en líneas de código principalmente. Por lo que puede presentar algunas deficiencias:

- Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizar las tareas.
- Se puede desviar de la realidad si indica mal el porcentaje de comentarios en las líneas de código.
- Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.
- La medición por líneas de código no es válida en la orientación a objetos.
- Utilizar esta herramienta puede resultar un poco complicada, en comparación con otros métodos (que también sólo estiman).

Teniendo en cuenta estas características sólo obtendremos una estimación aproximada del coste real del proyecto.

2.2 Aplicación del modelo

La función básica que utilizan los tres modelos (básico, intermedio y detallado) es:

$$E = a(Kl)^b \cdot m(X)$$

- **a** y **b**, son constantes con valores definidos en cada sub-modelo.
- **Kl**, son las líneas de código (en miles).
- **E**, es el resultado en salarios / mes y horas - persona.
- **m (X)**, es un multiplicador que depende de 15 atributos.

A la vez, cada sub-modelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- **Modo Orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía de unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles de líneas (medio).
- **Modo Semi-libre o Semi-encajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido o empotrado:** el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con la funcionalidad o técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

Vamos a emplear el tipo de modelo COCOMO básico, ya que es un modelo uni-variable estático.

Hay que añadir que este proyecto será considerado como software Orgánico, pues posee menos de 10.000 de líneas de código.

Podemos obtener una primera aproximación rápida del esfuerzo, y hacer uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Modo	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semi-Orgánico	3.00	1.12	2.50	0.35
Rígido	3.60	1.20	2.50	0.32

Tabla 1: Modelo COCOMO, Modelo básico

Estos valores serán usados para las fórmulas:

- Personas necesarias por mes para llevar adelante el proyecto:

$$(MM) = a(Kl^b)$$

- Tiempo de desarrollo del proyecto:

$$(TDEV) = c(MM^d)$$

- Personas necesarias para realizar el proyecto:

$$(CosteH) = \frac{MM}{TDEV}$$

- Costo total del proyecto:

$$(CosteM) = (CosteH) \cdot SM^1$$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

¹ *SM, Salario Medio de los programadores y analista.

2.3 Cálculos y resultados del modelo

La fórmula que aplicaremos es la del **tipo orgánico** y el método **básico**, puesto que estos son los modelos en los que cuadran este proyecto.

- Personas necesarias para finalizarlo en un mes:

$$MM = 2.40 * (10^{1.05}) = 27 \text{ personas}$$

- Tiempo de desarrollo:

$$TDEV = 2.50 * (27^{0.38}) = 8.74 \text{ meses}$$

- Personas necesarias en cada mes para finalizar en 8,74 meses:

$$CosteH = \frac{27}{8.74} = 3.08 \text{ personas por mes}$$

Tomando cómo aproximación del salario mínimo medio entre los programadores y analistas unos 1.200€ mensuales.

- Coste total del proyecto:

$$CosteM = 3.08 * 1200 = 3696 \text{ € mensuales}$$

- Coste Final del Proyecto:

$$CosteF = 8.74 * 3696 = \mathbf{32.303,04 \text{ €}}$$

Este coste final es sin incluir materiales, sólo el coste humano. Queremos recordar que el cálculo del 'coste' del proyecto en base a las líneas de código del producto final es muy relativo, ya que existen lenguajes con una sintaxis más potente y expresiva.

Además, añadimos la incertidumbre del uso de código reutilizado o repetido, es decir, si para repetir un conjunto de sentencias de código muy parecidas lo que se hace es copiar y/o modificar el código, según este método de estimación, nos habría supuesto un esfuerzo mayor, pero muchas veces puede ser más complicado y también más eficiente, crearnos con un mismo código, una función, que será llamada cuando sea necesaria con parámetros diferentes.

A lo largo de esta memoria se explicará en detalle nuestra filosofía seguida respecto a la repetición de código y otras decisiones de diseño que han influido en mayor o menor medida en el tamaño final de éste (número de líneas de código).

2.4 Planificación temporal

La **planificación temporal** consiste en la estimación del tiempo y esfuerzo durante las fases de desarrollo, con el fin de incrementar los niveles de servicio, calidad y coste.

Para ello, dividimos el ciclo de vida del proyecto en una serie de tareas a las cuales se les asignará las dos estimaciones siguientes:

- **Estimación inicial:** empleadas en los inicios del desarrollo. Son las menos exactas, pero se emplean como primera aproximación a la viabilidad del proyecto.
- **Estimación Final:** expresan la duración y el esfuerzo real empleado, cuyos valores se compararán para evaluar la exactitud de la estimación, empleando el Error Relativo de la estimación, RE.

Para hallar el error relativo aplicamos la formula estadística:

$$RE = \frac{(A - E)}{A}$$

En la cual,

- **A**, es el valor final real
- **E**, es la estimación inicial

A continuación, se muestra la estimación en horas de las principales tareas del proyecto:

Tareas	Estimación Inicial
Búsqueda de documentación	50 horas
Introducción	20 horas
Planificación	20 horas
Elicitación de requisitos	80 horas
Análisis de Requisitos	60 horas
Diseño del sistema	100 horas
Instalación del entorno de trabajo	10 horas
Implementación	200 horas
Pruebas	80 horas
Presentación	30 horas
TOTAL	650 horas

Tabla 2: Estimación inicial del proyecto

Aquí se describe el tiempo final que ha hecho falta para realizar cada tarea:

Tareas	Tiempo Final
Búsqueda de documentación	65 horas
Introducción	25 horas
Planificación	25 horas
Elicitación de requisitos	120 horas
Análisis de Requisitos	60 horas
Diseño del sistema	200 horas
Instalación del entorno de trabajo	50 horas
Implementación	300 horas
Pruebas	100 horas
Presentación	20 horas
TOTAL	965 horas

Tabla 3: Tiempo total empleado en el proyecto

Finalmente, aquí se muestra el error relativo en el cálculo de cada tarea:

Tareas	Error Relativo
Búsqueda de documentación	+ ~ 23,08 %
Introducción	+ 20 %
Planificación	+ 20 %
Elicitación de requisitos	+ ~33.33 %
Análisis de Requisitos	0 %
Diseño del sistema	+ 50 %
Instalación del entorno de trabajo	+ 80%
Implementación	+ ~33.33 %
Pruebas	+ 20 %
Presentación	- ~33.33 %
TOTAL	+ ~32,64 %

Tabla 4: Error Relativo cometido en la estimación inicial

CAPÍTULO 3: ELICITACIÓN DE REQUISITOS

3.1 Organizaciones y participantes

Organización	Escuela Politécnica Superior -Universidad Autónoma de Madrid
Dirección	Ciudad Universitaria de Cantoblanco, 28049 Madrid
Teléfono	+34 91 497 2222
Fax	
Comentarios	Email: informacion.eps@uam.es Web: http://www.uam.es/ss/Satellite/EscuelaPolitecnica/es/home.htm

Tabla 5: Descripción Organización 1

Organización	Facultad de Ingeniería, Universidad Nacional de Rosario
Dirección	Av. Pellegrini 250, S2000BTP Rosario, Santa Fe, Argentina
Teléfono	+54 341 480-2659
Fax	
Comentarios	Ninguno

Tabla 6: Descripción Organización 2

Organización	Organización de Desarrolladores
Dirección	Leonhardstraße 40A/2, 8010 Graz, Austria
Teléfono	+34 665817283
Teléfono Móvil	+43 68110556020
Comentarios	Ninguno

Tabla 7: Descripción Organización 3

Participante	Marc Bayón Benegas
Organización	<u>Organización de Desarrolladores</u>
Rol	Desarrollador
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 8: Participante 1 - Marc Bayón

Participante	Sergio Geninatti
Organización	<u>Facultad de Ingeniería, Universidad Nacional de Rosario</u>
Rol	Asesor
Es desarrollador	No
Es cliente	Sí
Es usuario	Sí
Comentarios	Ninguno

Tabla 9: Participante 2 - Sergio Geninatti

Participante	Eduardo Boemo Scalvinoni
Organización	<u>Escuela Politécnica Superior -Universidad Autónoma de Madrid</u>
Rol	Tutor
Es desarrollador	No
Es cliente	Sí
Es usuario	Sí
Comentarios	Ninguno

Tabla 10: Participante 3 - Eduardo Boemo

3.2 Objetivos del Proyecto

OBJ-0001	Gestión de la Seguridad
Versión	1.0 (05/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	El sistema deberá <i>proporcionar las herramientas necesarias para poder evitar accidentes a personas con discapacidad visual en el ámbito de las placas vitrocerámicas.</i>
Subobjetivos	<p>[OBJ-0006] Diseño de una llave maestra de desactivación remota y mecánica: El sistema deberá <i>incorporar una llave de desactivación en caso de emergencia, para ello, esta llave puede ser desactivada de manera remota o mecánica en caso de emergencia.</i></p> <p>[OBJ-0007] Aislamiento galvánico: El sistema deberá <i>estar prevenido y diseñado de manera que tenga separada galvánicamente la parte con acceso al usuario de la red de 220 v.</i></p> <p>[OBJ-0008] Técnicas de auto test y autodiagnóstico del sistema.: El sistema deberá <i>ser capaz de detectar algún problema en cualquier fase o estado del mismo, de manera que, en caso de anomalía y según la gravedad de la misma, pueda dejar de funcionar o avisar a al usuario con alguna señal de aviso.</i></p>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 11: OBJ-0001 - Gestión de la seguridad

OBJ-0002	Gestión de la Tele-lectura
Versión	1.0 (07/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	El sistema deberá <i>ser capaz de leer la información contenida en un display 7 segmentos de forma telemática, mediante el uso de algoritmos OCR y una cámara óptica.</i>
Subobjetivos	<p>[OBJ-0010] Gestión y tratamiento de imágenes: El sistema deberá <i>procesar cualquier tipo de imagen preparándola para el posterior procesamiento OCR.</i></p> <p>[OBJ-0011] Uso de algoritmos OCR.: El sistema deberá <i>ser capaz de interpretar la imagen de un display 7 segmentos, usando "Tesseract" como motor OCR de reconocimiento de caracteres.</i></p> <p>[OBJ-0014] Autoaprendizaje: El sistema deberá <i>ser capaz de auto aprender y auto entrenarse en la tarea de lectura OCR, mejorando el porcentaje de fiabilidad conforme se vaya incrementando su uso.</i></p>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 12: OBJ-0002 Gestión de la Tele-lectura

OBJ-0003	Gestión de la comunicación y tratamiento de datos
Versión	1.0 (07/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	El sistema deberá <i>ser capaz de comunicarse con un elemento externo vía bluetooth y gestionar el intercambio de mensajes.</i>
Subobjetivos	<p>[OBJ-0013] Sincronización y restablecimiento de las conexiones.: El sistema deberá <i>ser capaz de sincronizarse usando una comunicación inalámbrica. En caso de error en las comunicaciones, el sistema deberá ser capaz de restablecer la comunicación dado un número limitado de intentos. Tras alcanzarse el número de intentos, el sistema deberá abortar la comunicación.</i></p> <p>[OBJ-0016] Detección de proximidad a la fuente de calor y ayuda a la orientación: El sistema deberá <i>avisar al usuario, que está cerca de la fuente calor. Usar también esta tecnología para orientar la cámara del teléfono hacia el display 7 segmentos, solo en lugar de no disponer una cámara en la parte superior de la campana de la cocina.</i></p>
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 13: OBJ-0003 - Gestión de la comunicación y tratamiento de datos

OBJ-0004	Gestión de la Interactuación Usuario-Máquina
Versión	1.0 (07/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	El sistema deberá <i>proporcionar una interfaz de usuario apta para personas no videntes de tal manera que permita de forma sencilla el manejo de una vitrocerámica.</i>
Subobjetivos	<p>[OBJ-0005] Interfaz de manejo de la vitrocerámica: El sistema deberá <i>disponer de un elemento que gobierne el manejo de la vitrocerámica a partir de órdenes recibidas de una aplicación móvil. Este elemento deberá actuar de acuerdo a las órdenes recibidas.</i></p> <p>[OBJ-0009] Interfaz sonora para no videntes: El sistema deberá <i>disponer de una interfaz sonora usando una voz sintética que permita informar al usuario de estados del propio sistema o del resultado de acciones realizadas por el propio usuario. Esta parte debe implementarse en la aplicación móvil.</i></p> <p>[OBJ-0012] Interacción con gestos: El sistema deberá <i>usar "Android Gesture" para la recogida de órdenes por parte del usuario no vidente.</i></p>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 14: OBJ-0004 - Gestión de la Interactuación Usuario-Máquina

OBJ-0015	Gestión del Ahorro Energético
Versión	1.0 (09/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	El sistema deberá <i>ser capaz de ejecutarse en un dispositivo móvil, lo que prioriza el consumo de batería, de manera que sea mínimo en estado de reposo.</i>
Subobjetivos	<p>[OBJ-0017] Modos de funcionamiento: El sistema deberá <i>disponer de varios modos de uso de energía, a saber:</i></p> <ul style="list-style-type: none"> - <i>Screen Off: Funcionamiento cuando la pantalla está apagada, con un gasto energético menor.</i> - <i>Stand by: Funcionamiento a pleno rendimiento.</i>
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 15: OBJ-0015 - Gestión del ahorro energético

3.3 Requisitos

3.3.1 Requisitos de información

IRQ-0001	Vitrocerámica	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [CRQ-0001] Distancia entre máxima entre dispositivos • [UC-0023] Activación de la llave remota o mecánica • [UC-0003] Apagar Vitrocerámica • [UC-0005] Aumentar potencia • [UC-0007] Añadir tiempo de Cocción • [UC-0020] Eliminar tiempo de Cocción • [UC-0009] Emparejar vitrocerámica y aplicación móvil • [UC-0014] Desemparejar vitrocerámica y aplicación móvil • [UC-0013] Obtener información del dispositivo • [UC-0022] Desactivación por llave remota o mecánica • [UC-0002] Encender Vitrocerámica • [UC-0004] Disminuir potencia 	
Descripción	El sistema deberá almacenar la información correspondiente a la placa <i>vitrocerámica a controlar</i> . En concreto:	
Datos específicos	Número de serie identificativo único Modelo Estado Número del display 7 Segmentos Posición	
Tiempo de vida	Medio	Máximo
	3 mes(es)	6 mes(es)
Ocurrencias simultáneas	Medio	Máximo
	1	1
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	La vitrocerámica deberá disponer de la interfaz provista para poder volcar esta información en el aplicativo móvil.	

Tabla 16: IRQ-0001 Vitrocerámica

IRQ-0002	Gestos	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0012] Interacción con gestos • [UC-0002] Encender Vitrocerámica • [UC-0003] Apagar Vitrocerámica • [UC-0007] Añadir tiempo de Cocción • [UC-0020] Eliminar tiempo de Cocción • [UC-0004] Disminuir potencia 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>relación de gestos que el usuario puede realizar para controlar la placa vitrocerámica</i> . En concreto:	
Datos específicos	Encender la vitrocerámica Apagar la vitrocerámica Aumentar potencia Disminuir potencia Describir estado vía voz sintética Programar temporizador Resetear Temporizador	
Tiempo de vida	Medio	Máximo
	5 año(s)	10 año(s)
Ocurrencias simultáneas	Medio	Máximo
	7	14
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 17: IRQ-0002 Gestos

IRQ-0003	Sensores	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestión de la Seguridad • [UC-0008] Orientar Dispositivo Móvil • [UC-0016] Lanzar una alarma • [UC-0017] Añadir una configuración de ahorro energético • [UC-0018] Eliminar una configuración de ahorro energético • [UC-0019] Modificar una configuración de ahorro energético • [UC-0021] Obtener informes 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>sensores para detectar comportamientos anómalos en la vitrocerámica</i> . En concreto:	
Datos específicos	Identificador Tipo de sensor Medición	
Tiempo de vida	Medio	Máximo
	1 año(s)	2 año(s)
Ocurrencias simultáneas	Medio	Máximo
	1	4
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 18: IRQ-0003 Sensores

IRQ-0004	Actuadores	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestión de la Seguridad • [UC-0005] Aumentar potencia • [UC-0007] Añadir tiempo de Cocción • [UC-0020] Eliminar tiempo de Cocción • [UC-0010] Calibrar cámara de lectura • [UC-0017] Añadir una configuración de ahorro energético • [UC-0018] Eliminar una configuración de ahorro energético • [UC-0019] Modificar una configuración de ahorro energético • [UC-0021] Obtener informes • [UC-0004] Disminuir potencia • [UC-0002] Encender Vitrocerámica • [UC-0003] Apagar Vitrocerámica 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>actuadores del sistema</i> . En concreto:	
Datos específicos	Identificador Tipo Acción(es) a desempeñar	
Tiempo de vida	Medio	Máximo
	1 año(s)	2 año(s)
Ocurrencias simultáneas	Medio	Máximo
	1	5
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 19: IRQ-0004 Actuadores

IRQ-0005	Patrones numéricos	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0010] Gestión y tratamiento de imágenes • [OBJ-0011] Uso de algoritmos OCR. • [OBJ-0014] Autoaprendizaje • [UC-0001] Leer un dígito del display 7 segmentos de la vitrocerámica • [UC-0010] Calibrar cámara de lectura • [UC-0011] Añadir plantillas al motor OCR • [UC-0015] Eliminar plantillas al motor OCR • [UC-0012] Calibrar el software de mejora de imagen 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>plantillas de los dígitos a detectar a través de la tecnología OCR</i> . En concreto:	
Datos específicos	Imágenes de los números del 0 al 9 representados en un display 7 segmentos.	
Tiempo de vida	Medio	Máximo
	5 año(s)	10 año(s)
Ocurrencias simultáneas	Medio	Máximo
	10	10
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 20: IRQ-0005 Patrones numéricos

IRQ-0006	Estadísticas y Autoaprendizaje	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Autoaprendizaje • [UC-0001] Leer un dígito del display 7 segmentos de la vitrocerámica • [UC-0011] Añadir plantillas al motor OCR • [UC-0015] Eliminar plantillas al motor OCR • [UC-0012] Calibrar el software de mejora de imagen • [UC-0017] Añadir una configuración de ahorro energético • [UC-0018] Eliminar una configuración de ahorro energético • [UC-0019] Modificar una configuración de ahorro energético • [UC-0021] Obtener informes 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>fallos y aciertos en la lectura de un dígito</i> . En concreto:	
Datos específicos	Fallos en la lectura Aciertos en la lectura Patrón de píxeles mejorado Actividad de uso	
Tiempo de vida	Medio	Máximo
	1 año(s)	1 año(s)
Ocurrencias simultáneas	Medio	Máximo
	50	100
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 21: IRQ-0006 Estadísticas y Autoaprendizaje

IRQ-0007	Alertas	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0007] Aislamiento galvánico • [OBJ-0008] Técnicas de auto test y autodiagnóstico del sistema. • [OBJ-0001] Gestión de la Seguridad • [OBJ-0006] Diseño de una llave maestra de desactivación remota y mecánica • [UC-0023] Activación de la llave remota o mecánica • [UC-0016] Lanzar una alarma • [UC-0021] Obtener informes • [UC-0022] Desactivación por llave remota o mecánica 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>conjunto de alertas del sistema</i> . En concreto:	
Datos específicos	Fallo en el testeo de los sensores Comunicación Interrumpida entre aplicación móvil y vitrocerámica Temperatura anormalmente elevada Fallo en la comunicación bluetooth Fallo en la colocación del objeto a calentar Detección de sobre cocción Fallo en el autodiagnóstico Patrón OCR desconocido	
Tiempo de vida	Medio	Máximo
	6 mes(es)	1 año(s)
Ocurrencias simultáneas	Medio	Máximo
	1	10
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 22: IRQ-0007 Alertas

IRQ-0008	Estado	
Versión	1.0 (13/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0013] Sincronización y restablecimiento de las conexiones. • [OBJ-0016] Detección de proximidad a la fuente de calor y ayuda a la orientación • [OBJ-0006] Diseño de una llave maestra de desactivación remota y mecánica • [UC-0002] Encender Vitrocerámica • [UC-0003] Apagar Vitrocerámica • [UC-0005] Aumentar potencia • [UC-0007] Añadir tiempo de Cocción • [UC-0020] Eliminar tiempo de Cocción • [UC-0016] Lanzar una alarma • [UC-0017] Añadir una configuración de ahorro energético • [UC-0018] Eliminar una configuración de ahorro energético • [UC-0019] Modificar una configuración de ahorro energético • [UC-0021] Obtener informes • [UC-0004] Disminuir potencia 	
Descripción	El sistema deberá almacenar la información correspondiente al <i>estado del sistema</i> . En concreto:	
Datos específicos	estado anterior estado actual lista de posibles estados siguientes	
Tiempo de vida	Medio	Máximo
	15 segundo(s)	30 segundo(s)
Ocurrencias simultáneas	Medio	Máximo
	2	5
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 23: IRQ-0008 Estado

IRQ-0009	Mensajes de Voz Sintética	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0002] Gestión de la Tele-lectura • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0009] Interfaz sonora para no videntes • [UC-0002] Encender Vitrocerámica • [UC-0003] Apagar Vitrocerámica • [UC-0005] Aumentar potencia • [UC-0007] Añadir tiempo de Cocción • [UC-0020] Eliminar tiempo de Cocción • [UC-0016] Lanzar una alarma • [UC-0004] Disminuir potencia 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>relación de mensajes de voz sintética</i> . En concreto:	
Datos específicos	id Mensaje mensaje	
Tiempo de vida	Medio	Máximo
	6 mes(es)	1 año(s)
Ocurrencias simultáneas	Medio	Máximo
	1	1
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 24: IRQ-0009 Mensajes de Voz Sintética

IRQ-0010	Modos de Ahorro Energético	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0015] Gestión del Ahorro Energético • [OBJ-0017] Modos de funcionamiento • [UC-0018] Eliminar una configuración de ahorro energético • [UC-0019] Modificar una configuración de ahorro energético 	
Descripción	El sistema deberá almacenar la información correspondiente a <i>Configuración de ahorro energético</i> . En concreto:	
Datos específicos	Id Modo Ahorro parámetros modo activado	
Tiempo de vida	Medio	Máximo
	5 mes(es)	1 año(s)
Ocurrencias simultáneas	Medio	Máximo
	1	1
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 25: IRQ-0010 Modos de Ahorro Energético

3.4 Requisitos funcionales

3.4.1 Actores

ACT-0001	Usuario
Versión	1.0 (13/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	Este actor representa <i>el usuario final de aplicativo</i>
Comentarios	Se espera que el usuario sea una persona no vidente

Tabla 26: Actor 1 Usuario

ACT-0002	Superusuario
Versión	1.0 (18/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Descripción	Este actor representa <i>un usuario con privilegios de administración</i>
Comentarios	Este usuario tiene es un usuario vidente necesariamente y juega un papel de configurador del sistema.

Tabla 27: Actor 2 Superusuario

3.4.2 Diagramas de casos de uso

3.4.2.1 Representación del sistema

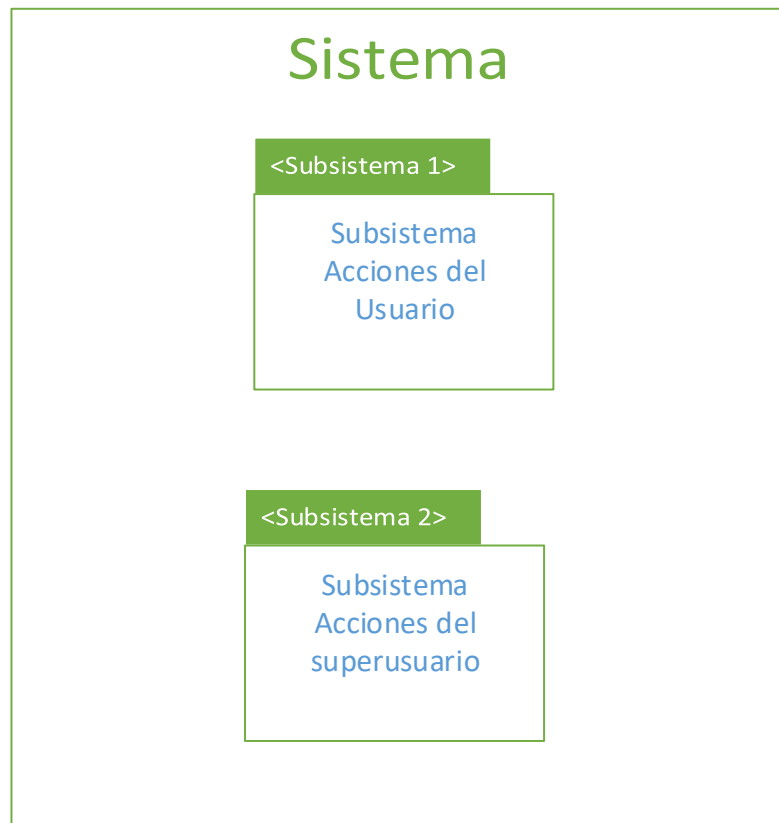


Ilustración 10: Diagrama del sistema y subsistemas principales

3.4.2.2 Subsistema de acciones del usuario

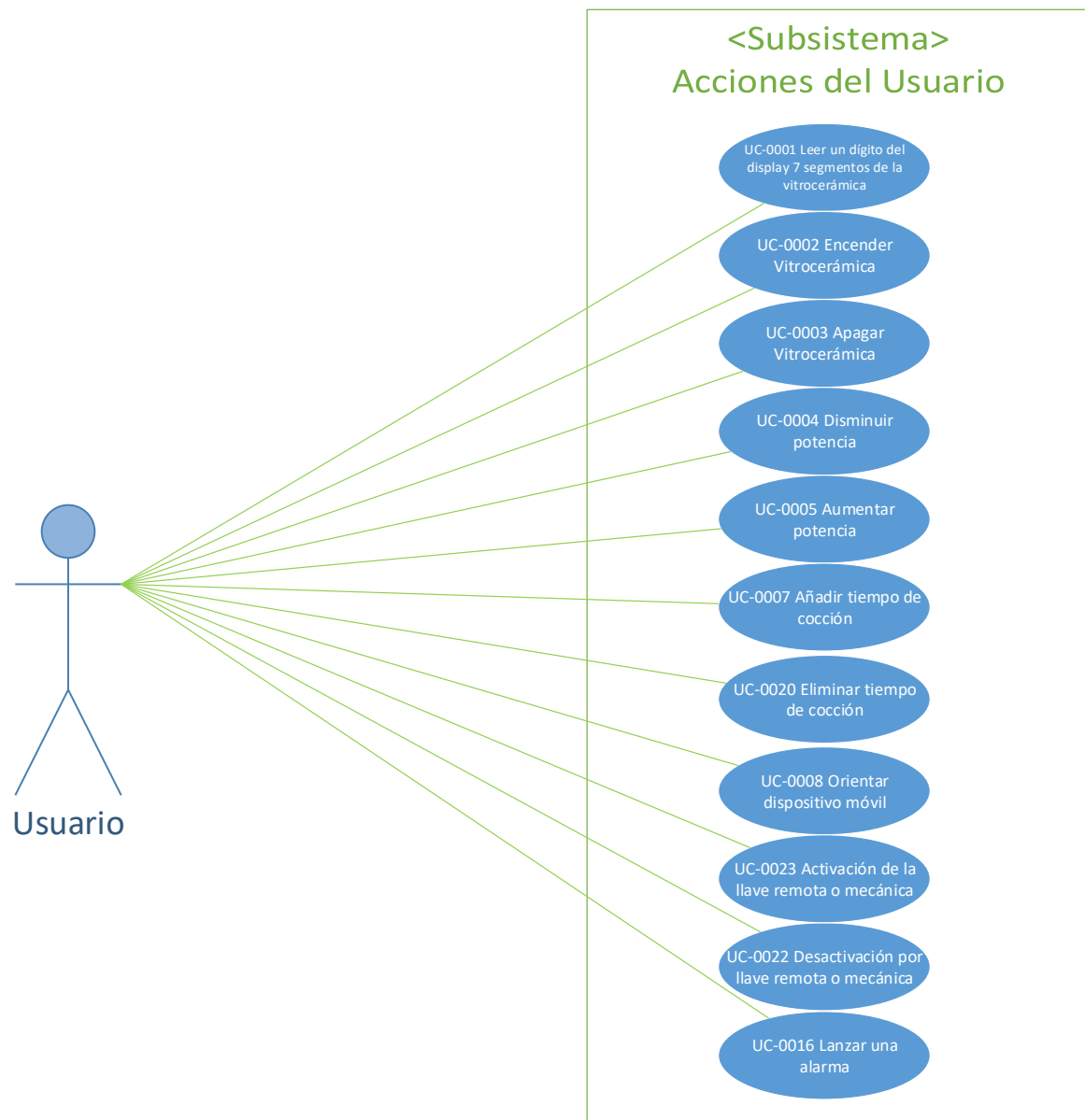


Ilustración 11: Subsistema de acciones del usuario

3.4.2.3 Subsistema de acciones del superusuario

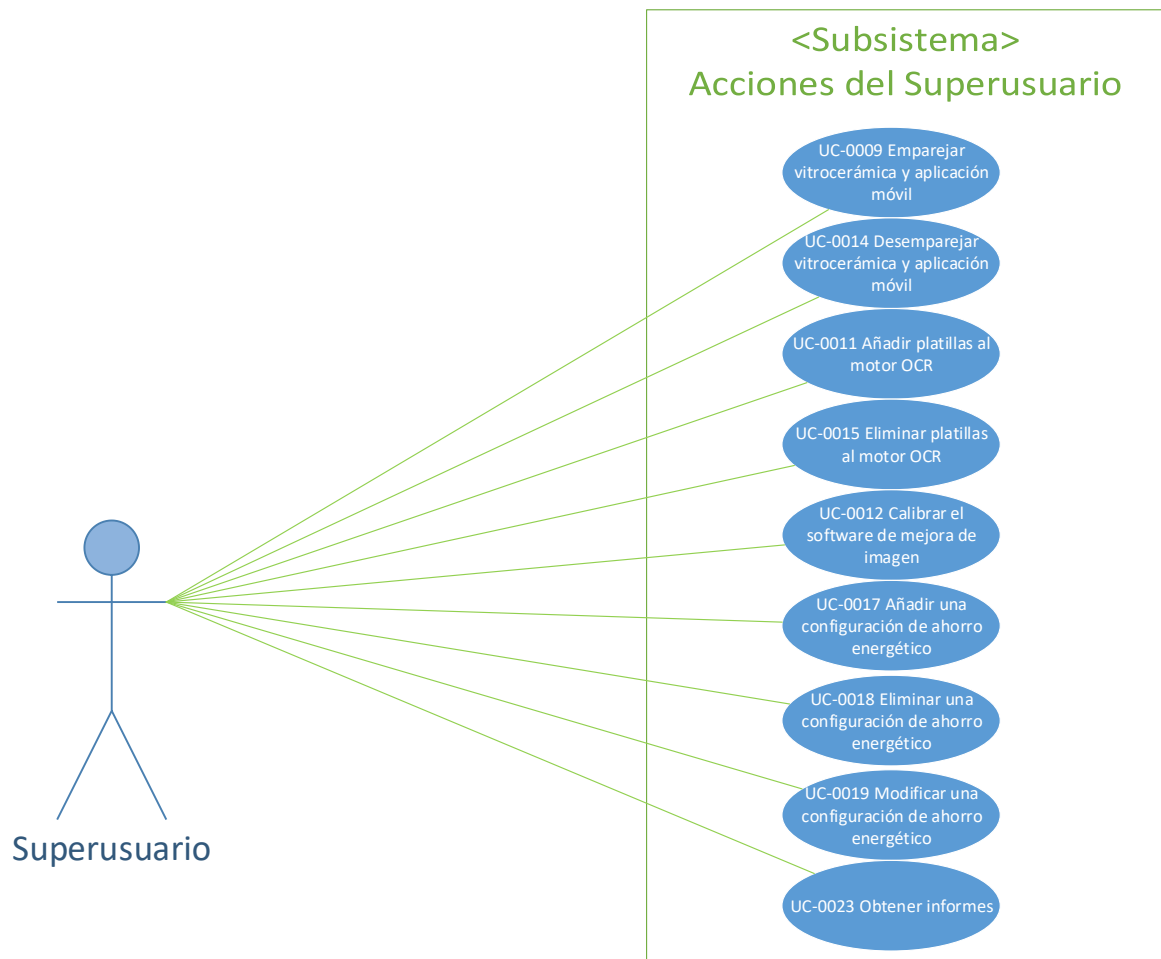


Ilustración 12: Subsistema de acciones del superusuario

3.4.3 Descripción de los casos de uso

UC-0001	Leer un dígito del display 7 segmentos de la vitrocerámica	
Versión	1.0 (16/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0002] Gestión de la Tele-lectura • [OBJ-0011] Uso de algoritmos OCR. • [OBJ-0010] Gestión y tratamiento de imágenes • [OBJ-0016] Detección de proximidad a la fuente de calor y ayuda a la orientación • [IRQ-0005] Patrones numéricos • [IRQ-0006] Estadísticas y Autoaprendizaje 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario toma una imagen con la cámara de fotos del teléfono o durante la realización de los siguientes casos de uso: [UC-0008] Orientar Dispositivo Móvil	
Precondición	La cámara del teléfono debe estar correctamente orientada hacia el display 7 segmentos.	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) selecciona en el aplicativo móvil que desea saber la información de la lectura del display 7 segmentos.
	2	El sistema verifica si existe una cámara preinstalada en el extractor de humos de la cocina.
	3	Si existe una cámara preinstalada en el extractor de humos, el sistema obtiene la imagen a través del dispositivo bluetooth
	4	El actor Usuario (ACT-0001) toma una imagen en la que aparezca el display 7 segmentos. (si se realiza con la cámara de fotos del teléfono móvil, debe orientar correctamente. Para ello, es necesario el uso de sensores de localización.)
Postcondición	Resultado de la comparación de los posibles patrones con la imagen tomada.	
Excepciones	Paso	Acción
	3	Si la comunicación ha fallado, el sistema indica al usuario que no ha sido posible usar la cámara externa, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	2	1 segundo(s)
	3	1 segundo(s)
Frecuencia esperada	1 veces por minuto(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 28: UC-0001 Leer un dígito del display 7 segmentos de la vitrocerámica

UC-0002	Encender Vitrocerámica	
Versión	1.0 (18/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0012] Interacción con gestos • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética • [IRQ-0002] Gestos • [IRQ-0001] Vitrocerámica • [IRQ-0004] Actuadores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario acciona el encendido de manera remota.</i>	
Precondición	La vitrocerámica debe estar apagada	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>dibuja un gesto sobre la pantalla del teléfono predefinido indicando que quiere encender la vitrocerámica.</i>
	2	<i>Si la vitrocerámica recibe la orden correctamente, el sistema la vitrocerámica pasa al estado encendida y comienza a calentar con la mínima temperatura.</i>
Postcondición	La vitrocerámica se encuentra encendida o en caso contrario se manda un mensaje de error al usuario.	
Excepciones	Paso	Acción
	2	<i>Si en caso de no recibir correctamente la orden, el sistema vuelve a reintentar restablecer la comunicación entre la vitrocerámica y la aplicación móvil, a continuación, este caso de uso queda sin efecto (Es posible reintentarlo unas pocas veces.)</i>
	2	<i>Si no existe o no está colocado correctamente el recipiente de cocción sobre la superficie a calentar, el sistema informa al usuario de que debe colocar éste de forma correcta, a continuación, este caso de uso continúa</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	4 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 29: UC-0002 Encender Vitrocerámica

UC-0003	Apagar Vitrocerámica	
Versión	1.0 (18/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0012] Interacción con gestos • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética • [IRQ-0001] Vitrocerámica • [IRQ-0002] Gestos • [IRQ-0004] Actuadores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario acciona el apagado de manera remota.</i>	
Precondición	La vitrocerámica debe estar encendida	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>dibuja un gesto sobre la pantalla del teléfono predefinido indicando que quiere apagar la vitrocerámica.</i>
	2	<i>Si la vitrocerámica recibe la orden correctamente, el sistema la vitrocerámica pasa al estado desconectando la superficie a calentar.</i>
Postcondición	La vitrocerámica se encuentra apagada o en caso contrario se manda un mensaje de error al usuario.	
Excepciones	Paso	Acción
	2	<i>Si en caso de no recibir correctamente la orden, el sistema vuelve a reintentar restablecer la comunicación entre la vitrocerámica y la aplicación móvil., a continuación, este caso de uso queda sin efecto (Es posible reintentarlo unas pocas veces.)</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	4 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 30: UC-0003 Apagar Vitrocerámica

UC-0004	Disminuir potencia	
Versión	1.0 (18/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0012] Interacción con gestos • [IRQ-0004] Actuadores • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética • [IRQ-0001] Vitrocerámica • [IRQ-0002] Gestos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea disminuir la potencia de cocción de forma remota</i>	
Precondición	La vitrocerámica debe hallarse encendida y debe existir un recipiente apto sobre la superficie correctamente colocado. La potencia debe ser > 0	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>dibuja un gesto sobre la pantalla del teléfono predefinido indicando que quiere aumentar la temperatura la vitrocerámica.</i>
	2	El sistema <i>aumenta la potencia en una unidad en la vitrocerámica.</i>
	3	Si <i>existe una instalación con cámara en la campana extractora de humos</i> , el sistema <i>realiza una foto con dicha cámara y realiza la lectura con OCR</i>
Postcondición	La potencia de la vitrocerámica se ha disminuir o en caso contrario se manda un mensaje de error informando al usuario.	
Excepciones	Paso	Acción
	3	Si <i>en caso de no coincidir el patrón con el siguiente número aumentado guardado en el sistema</i> , el sistema <i>el sistema alerta al usuario de que la potencia descrita por el display no coincide con la almacenada en el sistema</i> , a continuación, este caso de uso <i>continúa</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	10 veces por hora(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 31: UC-0004 Disminuir potencia

UC-0005	Aumentar potencia	
Versión	1.0 (18/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0012] Interacción con gestos • [IRQ-0004] Actuadores • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética • [IRQ-0001] Vitrocerámica • [IRQ-0002] Gestos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea aumentar la potencia de cocción de forma remota</i>	
Precondición	La vitrocerámica debe hallarse encendida y debe existir un recipiente apto sobre la superficie correctamente colocado. La potencia debe ser < MAX_POTENCIA	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>dibuja un gesto sobre la pantalla del teléfono predefinido indicando que quiere disminuir la temperatura la vitrocerámica.</i>
	2	El sistema <i>disminuye la potencia en una unidad en la vitrocerámica.</i>
	3	Si <i>existe una instalación con cámara en la campana extractora de humos</i> , el sistema <i>realiza una foto con dicha cámara y realiza la lectura con OCR</i>
Postcondición	La potencia de la vitrocerámica se ha aumentado, o en caso contrario se manda un mensaje de error informando al usuario.	
Excepciones	Paso	Acción
	3	Si <i>en caso de no coincidir el patrón con el anterior valor guardado en el sistema</i> , el sistema <i>el sistema alerta al usuario de que la potencia descrita por el display no coincide con la almacenada en el sistema</i> , a continuación, este caso de uso <i>continúa</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	10 veces por hora(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 32: UC-0005 Aumentar potencia

UC-0007	Añadir tiempo de Cocción	
Versión	1.0 (26/04/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0012] Interacción con gestos • [IRQ-0004] Actuadores • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética • [IRQ-0001] Vitrocerámica • [IRQ-0002] Gestos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiere configurar un temporizador para habilitar el tiempo cocción</i> .	
Precondición	La vitrocerámica está encendida.	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) desea activar una alarma temporizador o, en caso de que esté creada, el usuario desea añadir tiempo de cocción.
	2	El sistema le pide al usuario a través de la voz sintética que pronuncie o dibuje en la pantalla la hora que desea configurar o añadir
	3	El actor Usuario (ACT-0001) introduce los datos utilizando el canal de comunicación a convenir
	4	Si los datos introducidos han son correctos y correctamente interpretados, el sistema confirma los cambios y le pide al usuario que introduzca los dígitos correspondientes a los minutos.
	5	El actor Usuario (ACT-0001) introduce los minutos de la misma manera que ha introducido la hora
	6	Si los datos introducidos son correctos y bien interpretados, el sistema confirma los cambios y los notifica al usuario
Postcondición	El sistema habilita un contador de tiempo descendente.	
Excepciones	Paso	Acción
	4	Si el dato introducido no es correcto, el sistema vuelve a pedir que el usuario vuelva a introducir los datos de nuevo, a continuación, este caso de uso continúa
	6	Si el dato introducido no es correcto, el sistema vuelve a pedir que el usuario vuelva a introducir los datos de nuevo, a continuación, este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	5 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 33: UC-0007 Añadir tiempo de cocción

UC-0020	Eliminar tiempo de Cocción	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0005] Interfaz de manejo de la vitrocerámica • [OBJ-0012] Interacción con gestos • [IRQ-0004] Actuadores • [IRQ-0001] Vitrocerámica • [IRQ-0002] Gestos • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desee eliminar un temporizador habilitado para el tiempo cocción</i> .	
Precondición	La vitrocerámica está encendida y tiene un contador activo	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) desea eliminar una alarma temporizador.
	2	El sistema <i>le pide al usuario que seleccione con un gesto o pronuncie que contador quiere detener</i> .
	3	El actor Usuario (ACT-0001) introduce los datos utilizando el canal de comunicación a convenir
	4	Si los datos introducidos son correctos y bien interpretados, el sistema confirma los cambios y elimina el contador
Postcondición	El sistema elimina el contador	
Excepciones	Paso	Acción
	4	Si el dato introducido no es correcto, el sistema vuelve a pedir que el usuario vuelva a introducir los datos de nuevo, a continuación, este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	5 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 34: UC-0020 Eliminar tiempo de cocción

UC-0008	Orientar Dispositivo Móvil	
Versión	1.0 (13/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0016] Detección de proximidad a la fuente de calor y ayuda a la orientación • [IRQ-0003] Sensores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario necesita leer la información del display 7 segmentos con el teléfono móvil.</i>	
Precondición	El usuario activa la opción de lectura desde el dispositivo móvil	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Leer un dígito del display 7 segmentos de la vitrocerámica (UC-0001)
	2	El actor Usuario (ACT-0001) <i>demanda ayuda por parte del sistema para poder orientar el dispositivo.</i>
	3	El sistema <i>obtiene información del entorno y de la posición de la cámara y da órdenes sonoras al usuario acerca de la orientación y la proximidad.</i>
	4	El actor Usuario (ACT-0001) <i>obedece las indicaciones hasta que el dispositivo esté correctamente orientado</i>
	5	El sistema <i>notifica al usuario que el dispositivo está correctamente orientado.</i>
Postcondición		
Excepciones	Paso	Acción
	5	Si <i>no se encuentra la fuente de lectura</i> , el sistema <i>notifica al usuario que no ha sido posible encontrar la fuente de lectura.</i> , a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por hora(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 35: UC-0008 Orientar Dispositivo Móvil

UC-0009	Emparejar vitrocerámica y aplicación móvil	
Versión	1.0 (13/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0013] Sincronización y restablecimiento de las conexiones. • [IRQ-0001] Vitrocerámica • [CRQ-0001] Distancia entre máxima entre dispositivos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>configuración de la comunicación entre el dispositivo móvil y la vitrocerámica.</i>	
Precondición	La comunicación entre el dispositivo móvil y la vitrocerámica no está configurada	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea configurar la comunicación entre la vitrocerámica y el dispositivo móvil
	2	El sistema muestra al superusuario un formulario con las opciones de configuración
	3	El actor Superusuario (ACT-0002) rellena los campos con la configuración necesaria según demanda y circunstancias de la aplicación
	4	El sistema valida los campos y aplica la configuración dada por el superusuario.
Postcondición	La configuración queda configurada con los parámetros proporcionados por el superusuario.	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 36: UC-0009 Emparejar vitrocerámica y aplicación móvil

UC-0014	Desemparejar vitrocerámica y aplicación móvil	
Versión	2.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0013] Sincronización y restablecimiento de las conexiones. • [IRQ-0001] Vitrocerámica 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desee <i>eliminar información de emparejamiento entre el dispositivo móvil y la vitrocerámica.</i>	
Precondición	La vitrocerámica y el dispositivo móvil fueron previamente emparejados	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea <i>eliminar la comunicación entre la vitrocerámica y el dispositivo móvil</i>
	2	El sistema <i>muestra al superusuario un formulario con los dispositivos disponibles</i>
	3	El actor Superusuario (ACT-0002) <i>selecciona el dispositivo emparejado</i>
	4	El sistema <i>valida los campos y aplica la configuración dada por el superusuario.</i>
Postcondición	La vitrocerámica y el dispositivo quedan desemparejados.	
Excepciones	Paso	Acción
	4	Si <i>hay fallos en los parámetros de configuración</i> , el sistema <i>notifica al superusuario que existen fallos en los parámetros de configuración.</i> , a continuación, este caso de uso <i>queda sin efecto</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 37: UC-0014 Desemparejar vitrocerámica y aplicación móvil

UC-0010	Calibrar cámara de lectura	
Versión	1.0 (13/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0010] Gestión y tratamiento de imágenes • [OBJ-0002] Gestión de la Tele-lectura • [IRQ-0004] Actuadores • [IRQ-0005] Patrones numéricos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se precise <i>configuración de la comunicación entre el dispositivo móvil y la vitrocerámica</i> .	
Precondición	La cámara utilizada no está sacando imágenes de forma adecuada, aptas para el posterior procesamiento de imágenes.	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea calibrar la imagen que se obtiene de la cámara de fotos ;)
	2	El sistema muestra al superusuario un formulario con las opciones de configuración y una muestra de la imagen tomada. También preguntará al usuario si la imagen tomada debe ser desde una cámara instalada en una campana extractora de humos, o por el contrario debe usar la cámara que viene en el teléfono.
	3	El actor Superusuario (ACT-0002) rellena los campos con la configuración necesaria según demanda y circunstancias de la aplicación
	4	El sistema valida los campos y aplica la configuración dada por el superusuario.
Postcondición	La configuración queda configurada con los parámetros proporcionados por el superusuario.	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 38: UC-0010 Calibrar cámara de lectura

UC-0011	Añadir plantillas al motor OCR	
Versión	1.0 (13/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0011] Uso de algoritmos OCR. • [OBJ-0014] Autoaprendizaje • [IRQ-0005] Patrones numéricos • [IRQ-0006] Estadísticas y Autoaprendizaje 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desee <i>añadir plantillas nuevas a la base de imágenes del motor OCR</i>	
Precondición	Existe espacio disponible para plantillas	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea configurar las plantillas en el dispositivo OCR
	2	El sistema muestra al superusuario un formulario con las opciones de configuración
	3	El actor Superusuario (ACT-0002) rellena los campos con la configuración necesaria e introduce una serie de imágenes con las plantillas a reconocer en el motor OCR
	4	El sistema valida los campos y aplica la configuración dada por el superusuario.
Postcondición	Las plantillas quedan configuradas con los parámetros proporcionados por el superusuario.	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 39: UC-0011 Añadir plantillas al motor OCR

UC-0015	Eliminar plantillas del motor OCR	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0011] Uso de algoritmos OCR. • [OBJ-0014] Autoaprendizaje • [IRQ-0005] Patrones numéricos • [IRQ-0006] Estadísticas y Autoaprendizaje 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desea <i>eliminar plantillas nuevas a la base de imágenes del motor OCR</i>	
Precondición	La(s) plantilla(s) existen en la base de imágenes.	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea <i>eliminar las plantillas en el dispositivo OCR</i>
	2	El sistema <i>muestra al superusuario un formulario con las plantillas previamente configuradas</i>
	3	El actor Superusuario (ACT-0002) <i>selecciona las plantillas a eliminar</i>
	4	El sistema <i>valida los campos y aplica la configuración dada por el superusuario.</i>
Postcondición	Las plantillas quedan eliminadas de la base de imágenes del motor OCR	
Excepciones	Paso	Acción
	4	Si <i>hay fallos en los parámetros de configuración</i> , el sistema <i>notifica al superusuario que existen fallos en los parámetros de configuración.</i> , a continuación, este caso de uso <i>queda sin efecto</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 40: UC-0015 Eliminar plantillas del motor OCR

UC-0012	Calibrar el software de mejora de imagen	
Versión	1.0 (14/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0010] Gestión y tratamiento de imágenes • [IRQ-0005] Patrones numéricos • [IRQ-0006] Estadísticas y Autoaprendizaje 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desea realizar la <i>configuración del software de mejora de imagen y binarización con distintos valores de luz ambiente</i>	
Precondición	El software de mejora de imagen y binarización de la imagen no está configurado de forma correcta	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea configurar el software mejora de imagen y software de binarización.
	2	El sistema muestra al superusuario un formulario con las opciones de configuración
	3	El actor Superusuario (ACT-0002) rellena los campos con la configuración necesaria e introduce una serie de imágenes con las plantillas a reconocer en el motor OCR
	4	El sistema valida los campos y aplica la configuración dada por el superusuario.
Postcondición	La configuración queda configurada con los parámetros proporcionados por el superusuario.	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 41: UC-0012 Calibrar el software de mejora de imagen

UC-0013	Obtener información del dispositivo	
Versión	1.0 (14/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0003] Gestión de la comunicación y tratamiento de datos • [OBJ-0004] Gestión de la Interactuación Usuario-Máquina • [OBJ-0008] Técnicas de auto test y autodiagnóstico del sistema. • [OBJ-0012] Interacción con gestos • [IRQ-0001] Vitrocerámica 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiere saber en qué estado se encuentra la vitrocerámica</i>	
Precondición	ninguna	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) desea consultar el estado del sistema mediante un comando <i>gesture</i>
	2	El sistema <i>reproduce a través del altavoz del dispositivo móvil la información separada por elementos sonoros diferenciadores.</i>
Postcondición	El sistema informa al usuario del estado del sistema.	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	5 veces por hora(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 42: UC-0013 Obtener Información del dispositivo

UC-0016	Lanzar una alarma	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestión de la Seguridad • [OBJ-0007] Aislamiento galvánico • [OBJ-0008] Técnicas de auto test y autodiagnóstico del sistema. • [OBJ-0009] Interfaz sonora para no videntes • [IRQ-0003] Sensores • [IRQ-0007] Alertas • [IRQ-0008] Estado • [IRQ-0009] Mensajes de Voz Sintética 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>existe una anomalía en el sistema.</i>	
Precondición	Se ha detectado una anomalía en el sistema	
Secuencia normal	Paso	Acción
	1	El sistema <i>detecta una anomalía detectada por un sensor</i>
	2	El actor Usuario (ACT-0001) <i>recibe un aviso por parte del sistema avisando de la anomalía producida</i>
	3	<i>Si la incidencia es de tipo: exceso de calor, mal funcionamiento de la placa o incidencia eléctrica, el sistema deberá registrar la incidencia, desconectar el dispositivo y no permitir su uso.</i>
Postcondición	Se manda un mensaje de alerta al usuario y, según su gravedad, se desconecta el sistema para evitar daños mayores.	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por mes(es)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 43: UC-0016 Lanzar una alarma

UC-0017	Añadir una configuración de ahorro energético	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0015] Gestión del Ahorro Energético • [OBJ-0017] Modos de funcionamiento • [IRQ-0003] Sensores • [IRQ-0004] Actuadores • [IRQ-0006] Estadísticas y Autoaprendizaje • [IRQ-0008] Estado • [IRQ-0010] Modos de Ahorro Energético 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el superusuario desea añadir una configuración de ahorro energético.	
Precondición	El espacio para almacenar configuraciones de ahorro energético no debe estar lleno	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea añadir una configuración de ahorro energético en el sistema
	2	El sistema muestra al superusuario un formulario con las opciones de configuración
	3	El actor Superusuario (ACT-0002) rellena los campos con la configuración necesaria configurando la actividad de la conexión bluetooth, la duración de la actividad en la pantalla y demás parámetros que permiten un ahorro en la batería de los dispositivos del sistema.
	4	El sistema valida los campos y aplica la configuración dada por el superusuario.
Postcondición	La configuración queda guardada para ser usada por el usuario.	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 44: UC-0017 Añadir una configuración de ahorro energético

UC-0018	Eliminar una configuración de ahorro energético	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0015] Gestión del Ahorro Energético • [OBJ-0017] Modos de funcionamiento • [IRQ-0003] Sensores • [IRQ-0004] Actuadores • [IRQ-0006] Estadísticas y Autoaprendizaje • [IRQ-0008] Estado • [IRQ-0010] Modos de Ahorro Energético 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el superusuario desea eliminar una configuración de ahorro energético.	
Precondición	La configuración debe haber sido previamente creada	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea eliminar una configuración de ahorro energético en el sistema
	2	El sistema muestra al superusuario un formulario con las configuraciones previamente creadas.
	3	El actor Superusuario (ACT-0002) selecciona previamente la configuración a eliminar
	4	El sistema valida los campos, aplica la configuración dada por el superusuario y elimina la configuración
Postcondición	La configuración resulta eliminada	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 45: UC-0018 Eliminar una configuración de ahorro energético

UC-0019	Modificar una configuración de ahorro energético	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0015] Gestión del Ahorro Energético • [OBJ-0017] Modos de funcionamiento • [IRQ-0003] Sensores • [IRQ-0004] Actuadores • [IRQ-0006] Estadísticas y Autoaprendizaje • [IRQ-0008] Estado • [IRQ-0010] Modos de Ahorro Energético 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>superusuario</i> desea modificar una configuración de ahorro energético.	
Precondición	La configuración debe haber sido previamente creada	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) desea modificar una configuración de ahorro energético en el sistema
	2	El sistema muestra al superusuario un formulario con las configuraciones previamente creadas.
	3	El actor Superusuario (ACT-0002) selecciona la configuración a modificar
	4	El sistema valida los campos y aplica la modificación dada por el superusuario
Postcondición	La configuración resulta modificada	
Excepciones	Paso	Acción
	4	Si hay fallos en los parámetros de configuración, el sistema notifica al superusuario que existen fallos en los parámetros de configuración., a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 46: UC-0019 Modificar una configuración de ahorro energético

UC-0021	Obtener informes	
Versión	1.0 (19/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestión de la Seguridad • [OBJ-0008] Técnicas de auto test y autodiagnóstico del sistema. • [IRQ-0003] Sensores • [IRQ-0004] Actuadores • [IRQ-0006] Estadísticas y Autoaprendizaje • [IRQ-0007] Alertas • [IRQ-0008] Estado 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el superusuario desea obtener informes y logs acerca del funcionamiento del sistema</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Superusuario (ACT-0002) <i>solicita ver los informes generados por el sistema</i>
	2	El sistema <i>muestra al superusuario la lista de informes que son posibles generar.</i>
	3	El actor Superusuario (ACT-0002) <i>selecciona el informe o log que desea obtener</i>
	4	El sistema <i>muestra al usuario el informe seleccionado</i>
Postcondición	El sistema muestra los informes al usuario	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por mes(es)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 47: UC-0021 Obtener informes

UC-0022	Desactivación por llave remota o mecánica	
Versión	1.0 (20/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0007] Alertas • [IRQ-0001] Vitrocerámica • [OBJ-0001] Gestión de la Seguridad • [OBJ-0006] Diseño de una llave maestra de desactivación remota y mecánica 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario detecta una anomalía y desea desconectar rápidamente el electrodoméstico de la corriente eléctrica.</i>	
Precondición	La placa vitrocerámica está encendida y en cualquier estado	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) detecta una anomalía y desea desconectar la placa vitrocerámica. Para ello usa un comando gesture o acciona la llave mecánica.
	2	El sistema descarta toda acción y manda una señal al relé de corriente para desconectar la placa vitrocerámica de la corriente. Este relé queda desactivado hasta que vuelva a activarse.
Postcondición	La placa vitrocerámica queda desconectada de la corriente eléctrica	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por mes(es)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 48: UC-0022 Desactivación por llave remota o mecánica

UC-0023	Activación de la llave remota o mecánica	
Versión	1.0 (20/05/2016)	
Autores	Marc Bayón Benegas	
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0007] Alertas • [IRQ-0001] Vitrocerámica • [OBJ-0001] Gestión de la Seguridad • [OBJ-0006] Diseño de una llave maestra de desactivación remota y mecánica 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario, tras resolver una anomalía desea conectar de nuevo el aparato a la corriente eléctrica.</i>	
Precondición	La placa vitrocerámica está desconectada de la corriente eléctrica a través del relé	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>tras resolver la anomalía, desea desconectar la placa vitrocerámica. Para ello usa un comando gesture o acciona la llave mecánica.</i>
	2	El sistema <i>vuelve a accionar el relé para dar corriente a la placa nuevamente e inicia la placa en estado de reposo.</i>
Postcondición	La placa vitrocerámica está encendida y en estado de reposo.	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por mes(es)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 49: UC-0023 Activación de la llave remota o mecánica

3.5 Requisitos no funcionales

NFR-0001	Protección de datos
Versión	1.0 (13/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	El sistema deberá <i>aportar seguridad suficiente a la hora de guardar información sensible que pueda ser usada por terceros sin consentimiento final del usuario del aplicativo.</i>
Importancia	quedaría bien
Urgencia	puede esperar
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 50: NFR-0001 Protección de datos

NFR-0002	Reconexión
Versión	1.0 (20/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	El sistema deberá <i>reconectar una vez pierda la conexión entre la vitrocerámica y la aplicación móvil.</i>
Importancia	vital
Urgencia	hay presión
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 51: NFR-0002 Reconexión

NFR-0003	Autenticación
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	El sistema deberá <i>ofrecer un sistema básico de autenticación para el modo superusuario, de tal modo que la configuración solo pueda ser modificada por este Actor.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 52: NFR-0003 Autenticación

3.6 Reglas de negocio

CRQ-0001	Distancia entre máxima entre dispositivos
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>La distancia máxima de los dispositivos de la red no puede ser mayor a 30 metros</i>
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 53: CRQ-0001 Distancia máxima entre dispositivos

CRQ-0002	Límite de conexiones vinculadas
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>solo un actor puede estar vinculado con su aplicación móvil a una vitrocerámica a la vez.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 54: CRQ-0002 Límite de conexiones vinculadas

CRQ-0003	Derechos de superusuario para la configuración del sistema
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>Sólo el superusuario puede configurar el sistema. Un actor de tipo usuario no puede tener permisos para esta tarea.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 55: CRQ-0003 Derechos de superusuario para configuración del sistema

CRQ-0004	Tiempo máximo para reintentar la pérdida de una conexión
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>a partir de 1 minuto sin comunicación, la conexión se deberá abortar.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 56: CRQ-0004 Tiempo máximo para reintentar la pérdida de una conexión

CRQ-0005	Tiempo máximo de funcionamiento a pleno rendimiento
Versión	1.0 (26/04/2016)
Autores	Marc Bayón Benegas
Fuentes	Eduardo Boemo Scalvinoni Sergio Geninatti
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>El sistema estará a pleno rendimiento con las comunicaciones plenamente operativas mientras no exista un periodo de inactividad de más de 10 minutos. Tras ese periodo, el sistema entra en estado de reposo.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 57: CRQ-0005 Tiempo máximo de funcionamiento a pleno rendimiento

3.7 Matrices de Rastreabilidad

TRM-0001	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006	OBJ-0007	OBJ-0008	OBJ-0009	OBJ-0010	OBJ-0011	OBJ-0012	OBJ-0013	OBJ-0014	OBJ-0015	OBJ-0016	OBJ-0017
IRQ-0001	-	-	↑	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0002	-	-	-	↑	-	-	-	-	-	-	-	↑	-	-	-	-	-
IRQ-0003	↑	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0004	↑	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IRQ-0005	-	-	-	-	-	-	-	-	-	↑	↑	-	-	↑	-	-	-
IRQ-0006	-	-	-	-	-	-	-	-	-	-	-	-	-	↑	-	-	-
IRQ-0007	↑	-	-	-	-	↑	↑	↑	-	-	-	-	-	-	-	-	-
IRQ-0008	-	-	-	-	-	↑	-	-	-	-	-	-	↑	-	-	↑	-
IRQ-0009	-	↑	-	-	↑	-	-	-	↑	-	-	-	-	-	-	-	-
IRQ-0010	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↑	-	↑

Tabla 58: TRM-0001 Matriz relación objetivos del proyecto con Requisitos de información

TRM-0002	UC-0001	UC-0002	UC-0003	UC-0004	UC-0005	UC-0007	UC-0008	UC-0009	UC-0010	UC-0011	UC-0012	UC-0013	UC-0014	UC-0015	UC-0016	UC-0017	UC-0018	UC-0019	UC-0020	UC-0021	UC-0022	UC-0023
IRQ-0001	-	↑	↑	↑	↑	↑	-	↑	-	-	-	↑	↑	-	-	-	-	-	↑	-	↑	↑
IRQ-0002	-	↑	↑	↑	-	↑	-	-	-	-	-	-	-	-	-	-	-	-	↑	-	-	-
IRQ-0003	-	-	-	-	-	-	↑	-	-	-	-	-	-	-	↑	↑	↑	↑	-	↑	-	-
IRQ-0004	-	↑	↑	↑	↑	↑	-	-	↑	-	-	-	-	-	-	↑	↑	↑	↑	↑	-	-
IRQ-0005	↑	-	-	-	-	-	-	-	↑	↑	↑	-	-	↑	-	-	-	-	-	-	-	-
IRQ-0006	↑	-	-	-	-	-	-	-	-	↑	↑	-	-	↑	-	↑	↑	↑	-	↑	-	-
IRQ-0007	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↑	-	-	-	-	↑	↑	↑
IRQ-0008	-	↑	↑	↑	↑	↑	-	-	-	-	-	-	-	-	↑	↑	↑	↑	↑	↑	-	-
IRQ-0009	-	↑	↑	↑	↑	↑	-	-	-	-	-	-	-	-	↑	-	-	-	↑	-	-	-
IRQ-0010	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↑	↑	-	-	-	-

Tabla 59: TRM-0002 Matriz relación entre casos de uso y requisitos de información

TRM-0003	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006	OBJ-0007	OBJ-0008	OBJ-0009	OBJ-0010	OBJ-0011	OBJ-0012	OBJ-0013	OBJ-0014	OBJ-0015	OBJ-0016	OBJ-0017
UC-0001	-	↗	-	-	-	-	-	-	-	↗	↗	-	-	-	-	↗	-
UC-0002	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0003	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0004	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0005	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0007	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0008	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↗	-
UC-0009	-	-	↗	-	-	-	-	-	-	-	-	-	↗	-	-	-	-
UC-0010	-	↗	-	-	-	-	-	-	-	↗	-	-	-	-	-	-	-
UC-0011	-	-	-	-	-	-	-	-	-	-	↗	-	-	↗	-	-	-
UC-0012	-	-	-	-	-	-	-	-	-	↗	-	-	-	-	-	-	-
UC-0013	-	-	↗	↗	-	-	-	↗	-	-	-	↗	-	-	-	-	-
UC-0014	-	-	↗	-	-	-	-	-	-	-	-	-	↗	-	-	-	-
UC-0015	-	-	-	-	-	-	-	-	-	-	↗	-	-	↗	-	-	-
UC-0016	↗	-	-	-	-	-	↗	↗	↗	-	-	-	-	-	-	-	-
UC-0017	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↗	-	↗
UC-0018	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↗	-	↗
UC-0019	-	-	-	-	-	-	-	-	-	-	-	-	-	-	↗	-	↗
UC-0020	-	-	↗	↗	↗	-	-	-	-	-	-	↗	-	-	-	-	-
UC-0021	↗	-	-	-	-	-	-	↗	-	-	-	-	-	-	-	-	-
UC-0022	↗	-	-	-	-	↗	-	-	-	-	-	-	-	-	-	-	-
UC-0023	↗	-	-	-	-	↗	-	-	-	-	-	-	-	-	-	-	-

Tabla 60: TRM-0003 Matriz relación entre casos de uso y objetivos

CAPÍTULO 4: ARQUITECTURA DEL SISTEMA. DETALLES DE DISEÑO

4.1 Introducción

En este capítulo hablaremos de la arquitectura del sistema. Se entiende por arquitectura el conjunto de las partes ciertamente diferenciadas y divisibles en las que se podría dividir la aplicación. Antes de entrar en detalle, el siguiente diagrama muestra una visión global de cómo funcionaría todo el sistema:



Ilustración 13: Arquitectura del sistema

En la Ilustración 13 se muestra un diseño sencillo de la arquitectura que soportará el sistema. Dentro de la arquitectura se diferencian los siguientes componentes:



- **Vitrocerámica con display 7 Segmentos:**

Sistema de cocción en los que hay un vidrio entre la fuente de calor y el recipiente que se quiere calentar. Esta placa vitrocerámica debe venir equipada con un pequeño display 7 segmentos donde se pueda visualizar la temperatura o nivel de potencia de calor.



- **Unidad de control inteligente:**

Dispositivo computador de placa simple equipado con un microcontrolador y memoria de almacenamiento, conexión bluetooth y ethernet. También es posible que este dispositivo tenga instalado un sistema operativo de bajo nivel. En este caso este dispositivo será una Raspberry Pi 3 Modelo B.



- **Smartphone:**

Tipo de teléfono portable construido sobre una plataforma informática móvil, con mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional. Debe poseer conexión bluetooth y cámara de fotos (mínimo 5 MPX) para la lectura de OCR. El sistema Operativo debe ser un Android 4.1 o superior.

Para entrar en detalle en este diseño, dividiremos el sistema completo en dos subsistemas: Por un lado, la aplicación móvil dedicada al OCR y parte importante de este proyecto, y por otro lado la parte de comunicación bluetooth que incluye la programación del microcontrolador y los puertos de usuario GPIO que nos servirán para controlar el dispositivo a distancia.

4.2 Visión General

El diseño de este proyecto se divide en 2 subsistemas que se pasan a describir a continuación:

- Subsistema 1: OCR, captura, tratamiento y procesamiento de imágenes.
- Subsistema 2: Comunicación con unidad de control inteligente, gestión de órdenes y manejo de actuadores.

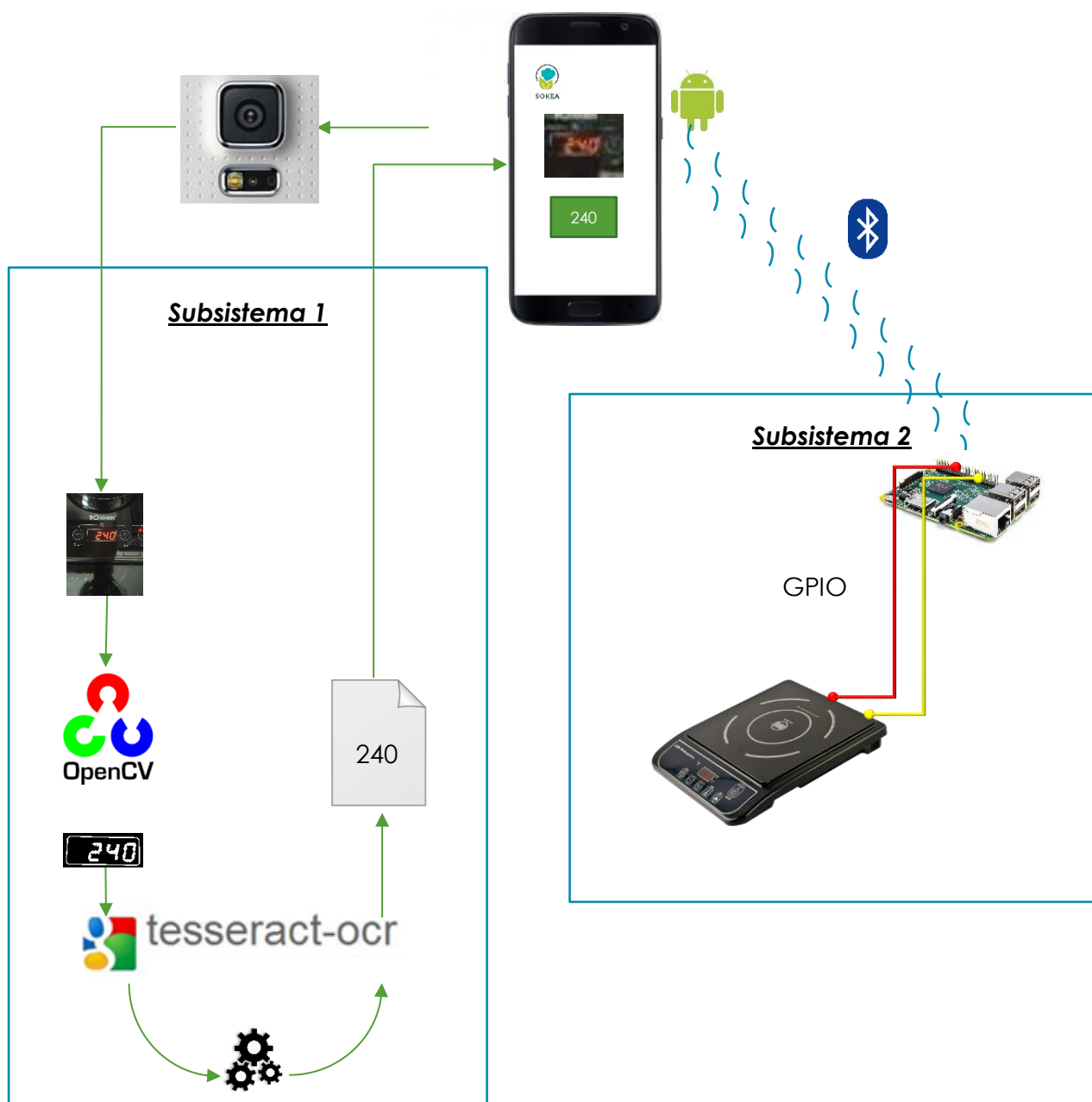


Ilustración 14: División de subsistemas.

4.2.1 Subsistema 1: Captura, tratamiento y procesamiento de imágenes.

En este subsistema se describe la parte que trata con el procesamiento de imágenes y extracción de metadatos. El procedimiento comienza con la captura con la cámara del teléfono una serie de imágenes por segundo para tratar de leer la cifra representada en el display 7 segmentos de la placa vitrocerámica.

El procedimiento es como se describe a continuación:

A través de la aplicación Android diseñada para este propósito, activaremos la captura de fotogramas a través de la cámara del teléfono. Se capturan fotogramas en formato PNG a color.

Por cada fotograma, iremos haciendo el siguiente procedimiento:

- 1) **Búsqueda de una región de interés:** La ROI (Region Of Interest) consiste en la búsqueda una subregión de la imagen donde es posible que se encuentre el dato que tenemos que leer. En este caso, buscamos un cuadrilátero con medidas superiores a 50 x 150 mm. Para detectar este cuadrilátero, utilizamos OpenCV [4] y la detección de contornos a partir de búsqueda de vértices y aproximación por mínimos cuadrados [5]. Una vez encontrada la región que satisfaga la propiedad, esta se recorta, para su posterior procesamiento en el siguiente punto. En caso contrario, se desestima el fotograma y se continua con el siguiente.



Ilustración 15: Detección automática de la región de interés por detección de cuadrados.

2) **Mejora de imagen:** Antes de pasar, a la fase de lectura e interpretación de los datos, necesitamos tratar la imagen para que sea tal y como se especifica en la documentación [6] del motor OCR utilizado (Tesseract en este caso). Para ello, usando de nuevo OpenCV, la imagen pasa por las fases siguientes:

- **Recorte y Binarización:** Una vez detectada el área de interés, se recorta la imagen junto por el área detectada. Luego, se realiza la binarización de la imagen. Dicho proceso consiste es pasar de una imagen a color (o escala de grises), a imagen en blanco y negro. Para ello se usa un umbral de binarización que es dependiente de la cantidad de luminosidad que tiene la captura de la imagen. Para ello se utiliza el umbral de binarización llamado el método Otsu [7], en honor a su creador Nobuyuki Otsu.

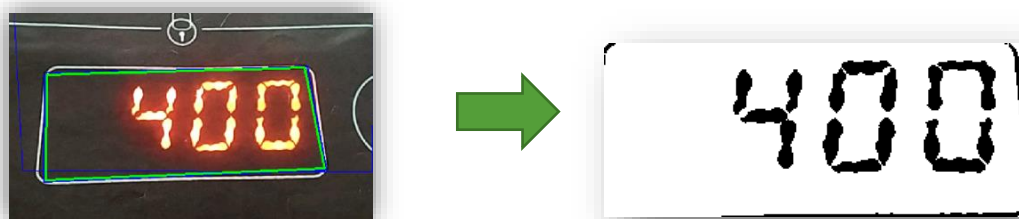


Ilustración 16: Recorte y Binarización método OTSU.

- **Suavizado:** El Suavizado consiste en un proceso filtrado de la imagen eliminando aquella información que no interesa de cara al reconocimiento. Para ello se utiliza un filtro de paso bajo. Una frecuencia alta en una imagen implica a un cambio brusco de intensidad entre blanco y negro (lo que se traduce pixeles aislados o ruido). Tras el paso de este filtro lo que se consigue es un suavizado de la imagen. OpenCV ofrece documentación [8] de cómo realizarlo de forma sencilla.

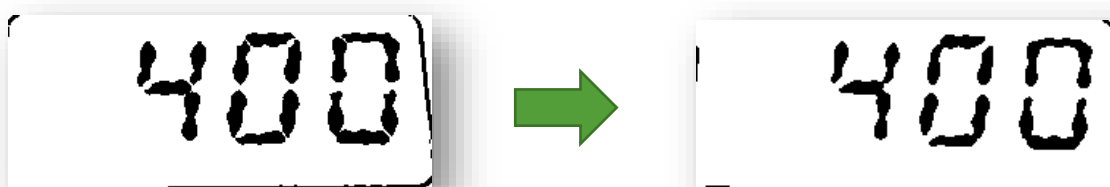


Ilustración 17: Suavizado de la imagen a partir de una imagen binaria.

- **Des-inclinación o “Deskew”:** Este procedimiento consiste en el enderezamiento de la imagen, causado por una posible toma inclinada. La des-inclinación logra detectar el ángulo [9] a corregir y aplicarlo acto seguido para conseguir una imagen enderezada. Como ya se he explicado, OpenCV trabaja muy bien con puntos, líneas y polinomios. Para detectar el ángulo, lo más sencillo es buscar líneas rectas. En caso de no tenerlas, es posible hacer un tratamiento para convertir cualquier bloque de texto en líneas horizontales.

Una vez obtenidas las líneas haz que calcular el ángulo en función de la vertical. Para ello se usa el arco tangente de cada línea y finalmente se computa la media de los ángulos. Con el ángulo resultante se aplica una translación de la imagen original a la imagen final girada, rellenando con pixeles blancos los posibles huecos que hayan podido quedar tras la translación.

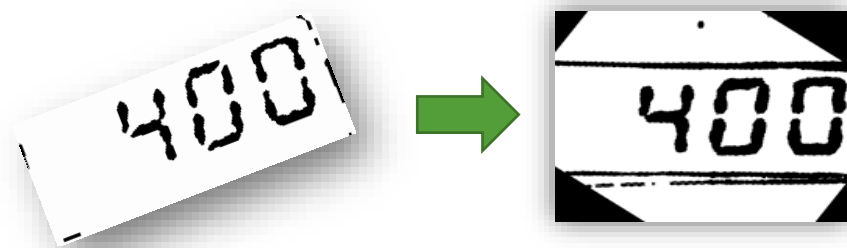


Ilustración 18: Ejemplo de deskew de una imagen realizada por la aplicación móvil.

- **Eliminación de líneas horizontales y verticales:** Debido a la necesidad de encontrar un cuadrilátero para la ROI, tenemos ahora el problema de que dichas líneas horizontales y verticales pueden dar lugar a confusión al motor OCR. Para ello, es necesario, como último paso, eliminarlas.

En este caso usamos un proceso de erosión con buscando líneas inferiores a 5 píxeles de grosor y las eliminaremos para dejar como resultado final una imagen sólo con la información que nos interesa para extraer de cara al motor OCR. Con esto acabaríamos el procesamiento del fotograma y tendríamos la imagen lista para la extracción de información.

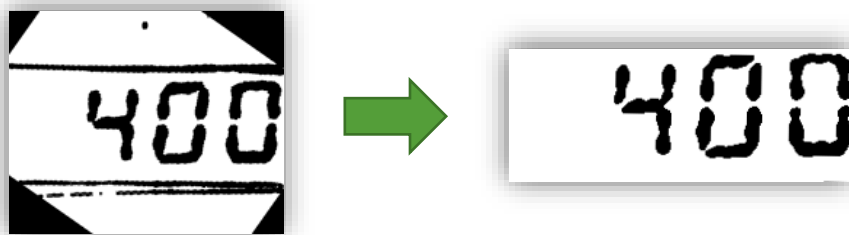


Ilustración 19: Eliminación de Líneas.

- 3) **Procesamiento de la imagen a través del motor OCR:** En este paso el motor OCR obtiene el fotograma previamente tratado por OpenCV y decide, usando su base de conocimiento previamente entrenada, el tipo de carácter que numérico que representan el conjunto de píxeles de la imagen. La parte complicada de este paso es el entrenamiento del motor para aprender a examinar el tipo de caracteres OCR 7 segmentos. Para ello dedicaremos un capítulo aparte que explica en profundidad como se realiza este procedimiento.

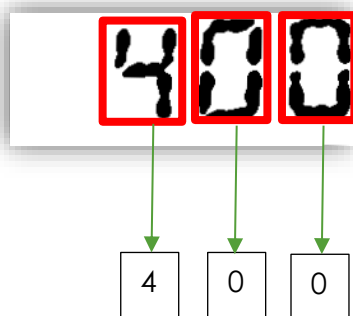


Ilustración 20: Procesamiento OCR

Una vez extraídos los metadatos de la imagen, la aplicación Android los muestra al usuario y acto seguido los reproduce a través del altavoz del teléfono inteligente. Con esto finaliza la función del subsistema 1.

4.2.2 Subsistema 2: Comunicación con unidad de control inteligente, gestión de órdenes y manejo de actuadores.

En este subsistema se describe la comunicación entre la aplicación móvil y el microcontrolador a través de **Bluetooth**. También se describirá la activación de los puertos GPIO para activar o desactivar funciones de la placa vitrocerámica.

La comunicación entre la aplicación móvil y la placa electrónica será a través de bluetooth usando el protocolo transporte de alto nivel L2CAP [10] cuyo funcionamiento se explicará en capítulos posteriores. Sobre L2CAP, el protocolo a usar elegido será "Binary protocol" [11] con control de errores para la capa de aplicación. Este protocolo consiste básicamente en enviar una serie de bytes con un control de errores al final de cada trama. La lista de comandos se describe en secciones posteriores: 4.5.2.

En la aplicación móvil todos los comandos a usar serán activables a través de gestos y a través de botones en la propia aplicación.

Por cada comando que se envía, se activa una entrada o salida en los puertos GPIO de la placa Raspberry Pi.

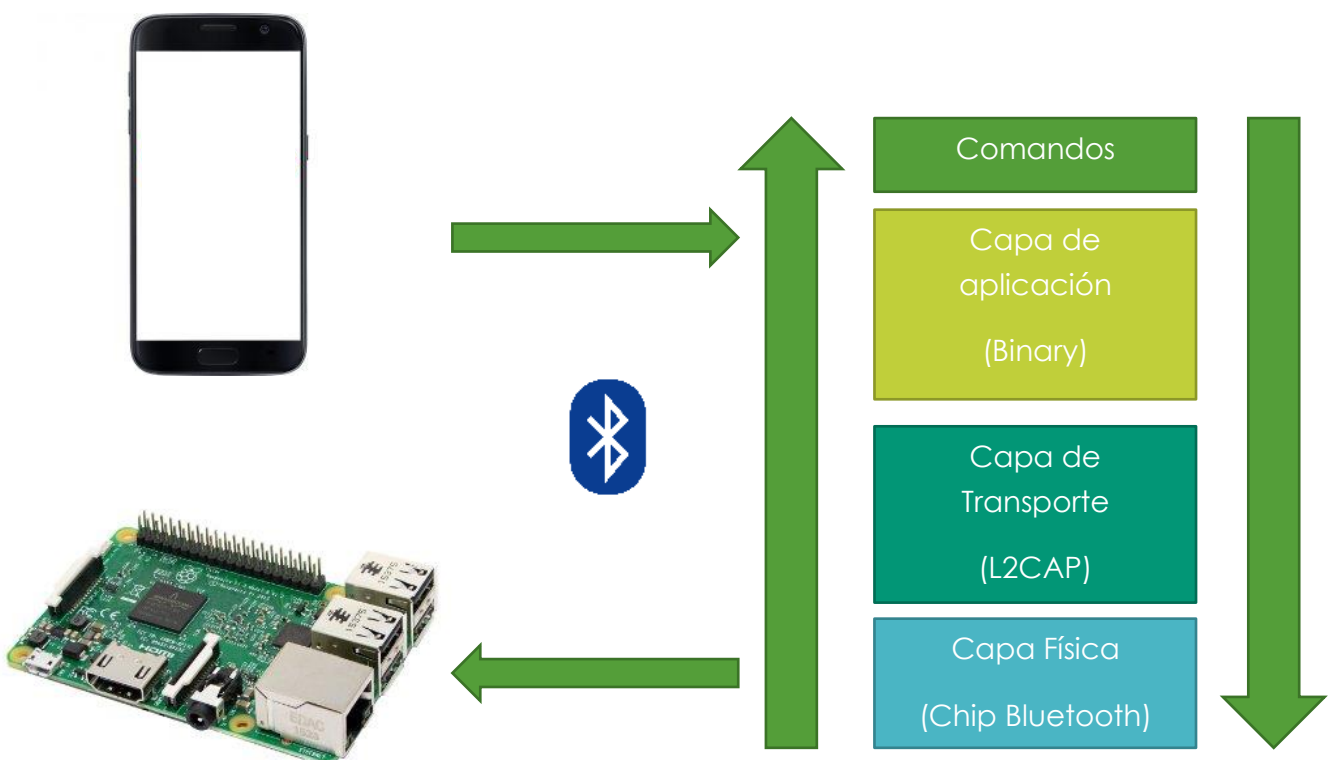


Ilustración 21: Protocolo de comunicación bluetooth para este TFM.

4.3 Definición de Tesseract© como motor OCR y fase de entrenamiento.

Tesseract es un motor OCR libre. Fue desarrollado originalmente por Hewlett Packard como software propietario entre 1985 y 1995. Tras diez años sin ningún desarrollo, fue liberado como código abierto en el año 2005 por Hewlett Packard y la Universidad de Nevada, Las Vegas. Esta herramienta está desarrollada actualmente por Google y distribuido bajo la licencia Apache, versión 2.0.

Tesseract está considerado como uno de los motores OCR libres con mayor precisión disponibles actualmente y sigue activo en desarrollo. A la hora de la redacción de este trabajo fin de máster, se usó la versión 3.04 lanzada en febrero de 2016. Tesseract es un motor OCR que permite reconocer actualmente 34 lenguas diferentes. Lo interesante de esta herramienta es que puede ser entrenada para reconocer cualquier otro lenguaje que no esté incluido en su suite de origen. En este caso este motor OCR fue elegido entre otros por desempeñar esta capacidad de entrenamiento y puede ser aprovechado, entre otras cosas, para leer Displays 7 segmentos.

4.3.1 Descripción de la fase de entrenamiento.

A pesar de que Tesseract es multiplataforma, únicamente es posible entrenarlo en un sistema operativo de tipo Linux. Para ello en este TFM se usó una distribución de Linux Ubuntu versión 16.04 como entorno de entrenamiento.

La fase entrenamiento de este motor OCR no es un proceso trivial, se necesitan varios elementos que pasamos a describir a continuación.

4.3.1.1 Preparación del sistema operativo.

Como se ha comentado en este apartado, necesitamos preparar un sistema operativo tipo Linux para poder realizar el entrenamiento de Tesseract. Para ello es recomendable usar una partición en un disco físico o en una máquina virtual dedicada. Es recomendable no utilizar un sistema Linux usado previamente para tratamiento de imágenes debido que existen conflictos con la librería para ficheros PNG que instala por defecto, por ejemplo, el software de tratamiento de imágenes GIMP.

4.3.1.1.1 Instalar Leptonica

Leptonica [12] es un software de código abierto desarrollado por Dan Bloomberg, y la IEEE para uso académico. Es especialmente útil para el procesamiento y análisis de imágenes. Tesseract usa leptonica para tratar la imagen antes de procesarla. Es especialmente útil para detectar monoespaciados entre los caracteres a reconocer. Para poder entrenar Tesseract es necesario su instalación que detallamos a continuación:

- 1) Instalar las siguientes librerías como prerequisite:

```
sudo apt-get install libjpeg-dev libtiff5-dev libpng16-dev  
libgif-dev libwebp-dev libopenjp2-7-dev zlib1g-dev
```

- 2) Nos descargamos el código fuente de Leptonica desde el siguiente repositorio: <http://www.leptonica.com/download.html>
- 3) Configuramos, compilamos y comprobamos que el proceso ha salido correctamente, ejecutando los siguientes comandos:

```
./configure  
make  
sudo make install  
sudo make check
```

Con esto tendremos instalado y configurado Leptonica.

4.3.1.1.2 Instalar Tesseract

Para instalar Tesseract en Linux es sencillo. Tan solo hay que ejecutar este comando para instalarlo desde el repositorio:

```
sudo apt-get install tesseract-ocr tesseract-ocr-vie
```

Con esto tendremos configurado la máquina para ejecutar Tesseract y Leptonica.

4.3.1.2 Descripción del procedimiento de entrenamiento

Una vez instalado Tesseract y leptónica ya tenemos la base para poder a empezar a entrenar. Este motor OCR requiere como elemento final un fichero con extensión "traineddata" con el nuevo lenguaje a leer. Para generar este fichero se requiere la ejecución de varios comandos en Tesseract, así como la disposición del material de entrenamiento que se enumeran a continuación:

- 1) Es necesario tomar un cierto número de muestras de cada fuente a reconocer. Cada muestra debe ser en binario una resolución de 300 pixeles por pulgada (PPP o DPI). En cuanto a la calidad de las muestras, se recomienda algo cercano a la calidad media con la que se van a capturar. Si entrenamos Tesseract con muestras con calidad superior a la esperada, lo estaremos sobreentrenando, no consiguiendo resultados deseados. Si, por el contrario, seleccionados muestras con una calidad inferior a la captura real, tampoco obtendremos resultados buenos. Hay que tener siempre en cuenta que Tesseract lee pixeles y no texto. En base a como se haya entrenado, nos lanzará un resultado u otro. En nuestro caso, la muestra debe ser lo más cercana a la realidad, pues la idea es que lo pixeles que representan a un dígito como puede ser un cero, estén siempre en torno al mismo lugar desde un punto de vista estadístico, aunque la calidad a nuestros ojos no sea la óptima. En caso de que por ejemplo estemos tratando de leer caracteres sobre un documento digital como puede ser un PDF, la calidad de las muestras de entrenamiento sí que debe ser óptima. Para ello es recomendable preinstalar una fuente de texto en el sistema con la grafía similar a la que se quiere y generar la imagen de cada carácter.

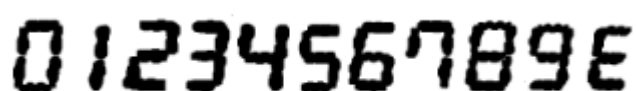


Ilustración 22: Muestras recogidas por la cámara del teléfono con calidad media.

- 2) Finalmente se debe componer un fichero con extensión .TIFF sin compresión como este con ciertas repeticiones de cada serie por cada línea de texto:

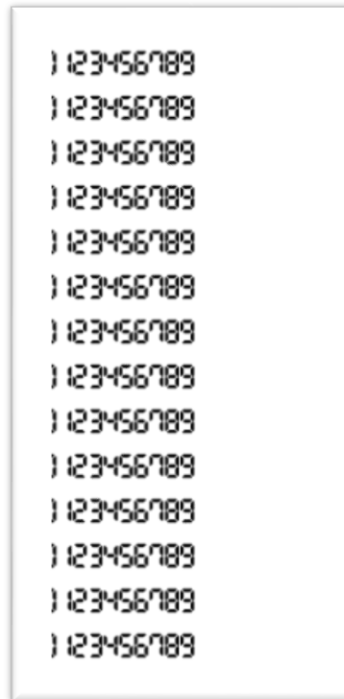


Ilustración 23: Fichero de entrenamiento de tesseract.

El fichero debe seguir una nomenclatura específica como se especifica a continuación: [lang].[fontname].exp[num].tif

- 3) Una vez tenemos la muestra correctamente generada, es necesario decirle a Tesseract en qué posición se encuentra cada carácter en la muestra. Tesseract genera este fichero a partir de la imagen TIF generada en el paso anterior con tan solo ejecutar el siguiente comando en la consola de Linux:

```
tesseract 7seg.XXXX.exp0.tif 7seg.XXXX.exp0 batch.nochoop makebox
.
.
.
tesseract 7seg.XXXX.expN.tif 7seg.XXXX.expN batch.nochoop makebox
```

Una vez finalizado, Tesseract debería dejar en la misma ruta que la imagen un fichero con extensión ".box". Para visualizar el resultado se recomienda usar la herramienta jTessBoxEditor (añadir fuente) <http://vietocr.sourceforge.net/training.html>

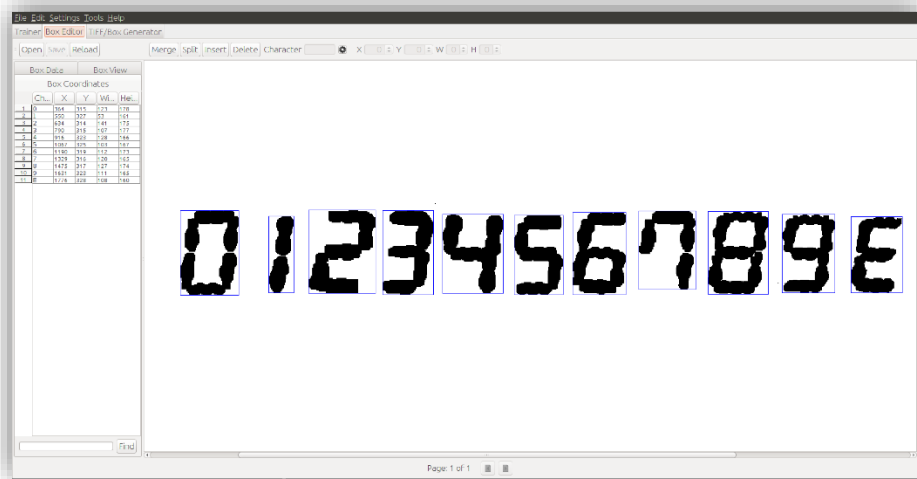


Ilustración 24: Visión del programa generador de cajas vietOCR

Una vez abrimos la aplicación, simplemente corregimos y ajustamos los valores devueltos por Tesseract y guardamos el fichero.

- 4) El siguiente paso es generar el fichero de entrenamiento por cada fichero tif/box que hayamos generado. Para ello ejecutamos:

```
tesseract 7seg.XXXX.exp0.tif 7seg.XXXX.exp0 nobatch box.train
.
.
.
tesseract 7seg.XXXX.expN.tif 7seg.XXXX.expN.box nobatch box.train
```

- 5) En este paso generaremos el fichero "unichart". Tesseract genera este fichero automáticamente a partir de los ficheros ".box" que hayamos generado ejecutando el siguiente comando:

```
unicharset_extractor 7seg.XXXX.exp0.box 7seg.XXXX.exp1.box...
7seg.XXXX.expN.box
```

- 6) Abrimos un fichero en blanco y generamos el fichero 7seg.font_properties. Este fichero solo debe contener la siguiente información:

```
segment7 0 0 0 0 0
```

En primer lugar, se coloca el nombre la fuente, y luego cinco dígitos que son bits para describir la fuente. Los dígitos significan en este orden: <italic>, <bold>, <fixed>, <serif> y <fraktur> y obedecen a banderas con valor 0 o 1, de tal manera que 1 significará que la propiedad está activada y 0 desactivada. Para nuestro ejemplo asumimos que nuestra fuente carece de ninguna de esas propiedades. Acto seguido guardamos el fichero como "7seg.font_properties" y concluimos con este paso.

- 7) Acto seguido hace el "shapeclustering" de los ficheros de entrenamiento, junto con el fichero "unicharset" y nuestro "font_properties" para crear la tabla de formas de la fuente con el siguiente comando:

```
shapeclustering -F font_properties -U unicharset 7seg.XXXX.exp0.tr  
7seg.XXXX.exp1.tr ... 7seg.XXXX.expN.tr
```

- 8) Usamos el comando "mftraining" y "cntraining" para crear una regla de sensibilidad de detección de los prototipos.

```
mftraining -F font_properties -U unicharset -O tla.unicharset  
tla.test_font.exp0.tr tla.test_font.exp1.tr ... tla.test_font.exp31.tr  
  
cntraining      tla.test_font.exp0.tr      tla.test_font.exp1.tr      ...  
tla.test_font.exp31.tr
```

- 9) Renombrar los ficheros *shapetable*, *normproto*, *inttemp* y *pfmtable*.

```
move inttemp tla.inttemp  
move normproto tla.normproto  
move pfmtable tla.pfmtable  
move shapetable tla.shapetable
```

- 10) Combinar los ficheros para obtener el resultado final.

```
combine_tessdata tla.
```

Con estos 10 pasos tenemos el fichero de entrenamiento que debemos guardar para instalarlo en la aplicación móvil para que lo lea Tesseract.

4.4 Arquitectura de la aplicación móvil

Para la aplicación móvil se ha elegido la plataforma Android para construir la aplicación. La interfaz de usuario ha sido pensada para ser manejada por personas con discapacidad visual limitada o nula, con lo que se trabajará bastante con comandos por voz y por altavoz. Pero antes de entrar en detalle acerca de la interfaz visual, nos centraremos en el “back-end” del aplicativo y la comunicación entre capas de software hasta llegar a visualizar el dato en la interfaz.

4.4.1 Back-end de la aplicación.

Nos referimos a back-end como al conjunto de capas de software que no pertenece propiamente a la interfaz de usuario y que se ejecutan internamente en la aplicación de formas transparente. Esta es la arquitectura del backend.

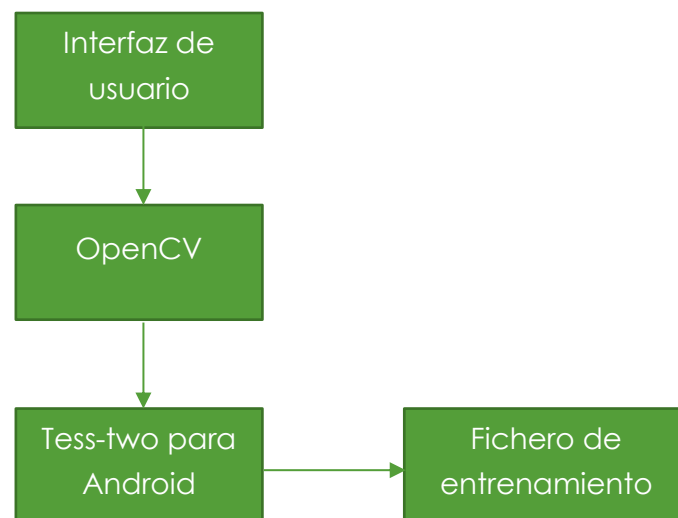
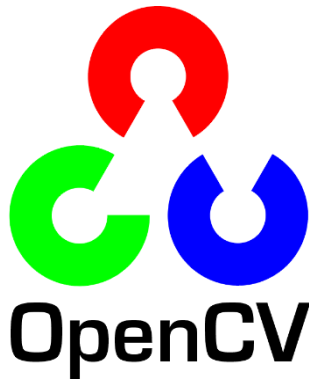


Ilustración 25: Backend de la aplicación

4.4.1.1 OpenCV



OpenCV (<https://es.wikipedia.org/wiki/OpenCV>) es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV juega un papel crucial en este proyecto. Es el encargado de obtener la región de interés, y de mejorar la imagen y prepararla para el OCR. Se utiliza a modo de API en la aplicación android. Existe una librería de OpenCV preparada para Android que nos facilita su uso e integración.

4.4.1.2 "Tess-two" Wrapper Tesseract para Android.

Tess-two es el proyecto desarrollado por Robert Theis [13] que integra Tesseract para poder ejecutarse bajo una aplicación Android. El proyecto compila las librerías nativas de Tesseract bajo un proyecto Android en Java usando JNI [14].

Previo a su uso, Tess-two debe ser compilado completamente para generar la librería java que posteriormente usaremos en nuestra aplicación. En el capítulo siguiente explicamos cómo instalar el entorno completo incluyendo esta parte bajo el IDE eclipse.

Para terminar, comentar que tess-two lee nuestro fichero de entrenamiento previamente creado en la sección 4.3.1.2 de un directorio denominado tessdata que deberá estar colocado en el directorio raíz de nuestra aplicación.

Para usar Tess-Two, es necesario compilarlo previamente. Para ello es necesario usar Android NDK. Explicaremos en capítulos posteriores cómo se puede instalar Tess-two para poder usarlo en un entorno de desarrollo: 5.8.1

4.4.2 Front-End de la aplicación

Nos referimos con front-end a la parte de software que interactúa con los usuarios finales de la aplicación. En este punto se explicará las diferentes pantallas que podemos encontrar. En primer lugar y en términos de usabilidad, la aplicación Android está diseñada para personas con visión muy reducida o nula. Para acceder a la aplicación, google facilita una interfaz por voz para ejecutar aplicaciones. La aplicación que google usa se denomina Google TalkBack y está disponible para descarga en Google Play y normalmente viene preinstalada en dispositivos Android.

En caso de no poder contar con la característica de Google Talkback, he diseñado un widget suficientemente grande que se puede poner en la pantalla principal del teléfono y ejecutar la aplicación haciendo las veces de icono de la aplicación.

Se ha implementado una funcionalidad que permite controlar la aplicación con gestos. La lista de comandos que la aplicación soporta por gestos son los siguientes:



Modo OC, para lectura del display 7 segmentos (se abre otra pantalla)



Conectarse a la vitrocerámica



Desconectarse de la vitrocerámica



Empezar la cocción



Finalizar la cocción



Aumentar potencia de cocción



Disminuir potencia de cocción

A continuación, se muestra el menú de navegabilidad del aplicativo diseñado para esta versión donde se pueden diferenciar los subsistemas definidos en este capítulo, la aplicación se ha pensado hacerla lo más sencilla posible por lo que solo contendrá 4 pantallas que son las siguientes:

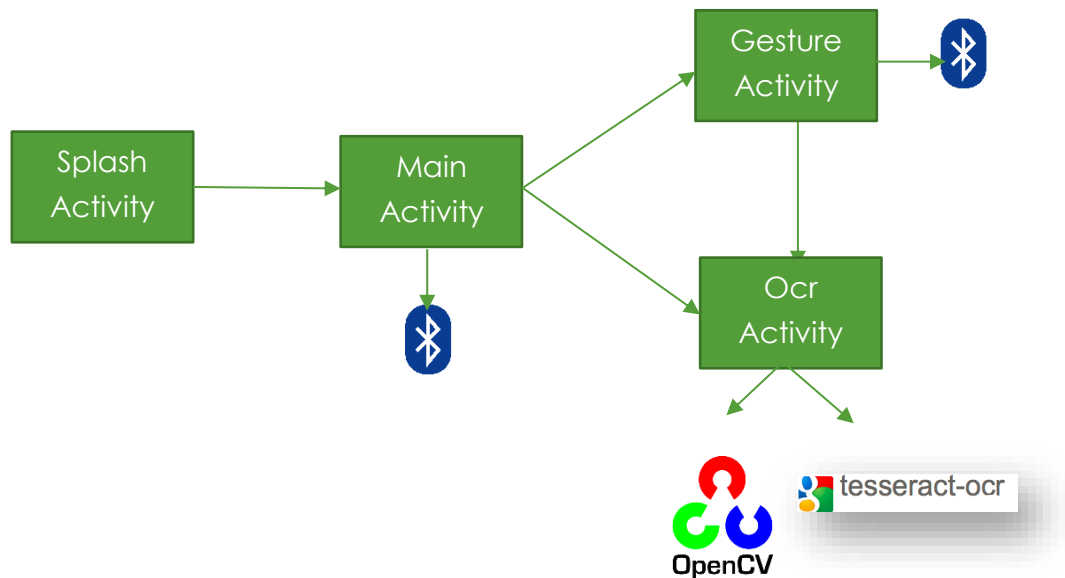


Ilustración 26: Esquema de navegabilidad de la aplicación.

La aplicación en principio se pensó solo para leer únicamente displays 7 segmentos. Más adelante se le añadió la funcionalidad de la conectividad bluetooth, lo que requirió rediseñar la interfaz de usuario.

Para el diseño previo de la interfaz de usuario se utilizaron varios modelos *wireframes* [15]. Un *wireframe* es una guía visual que representa el esqueleto o estructura visual de una aplicación web o aplicación móvil. Son útiles a la hora de implementar diseños de interfaces gráficas, ya que nos hacen tener una primera idea de la navegabilidad de la aplicación y también de la parte gráfica o artística esperada.

A continuación, se presente el *wireframe* final para el diseño de la interfaz:

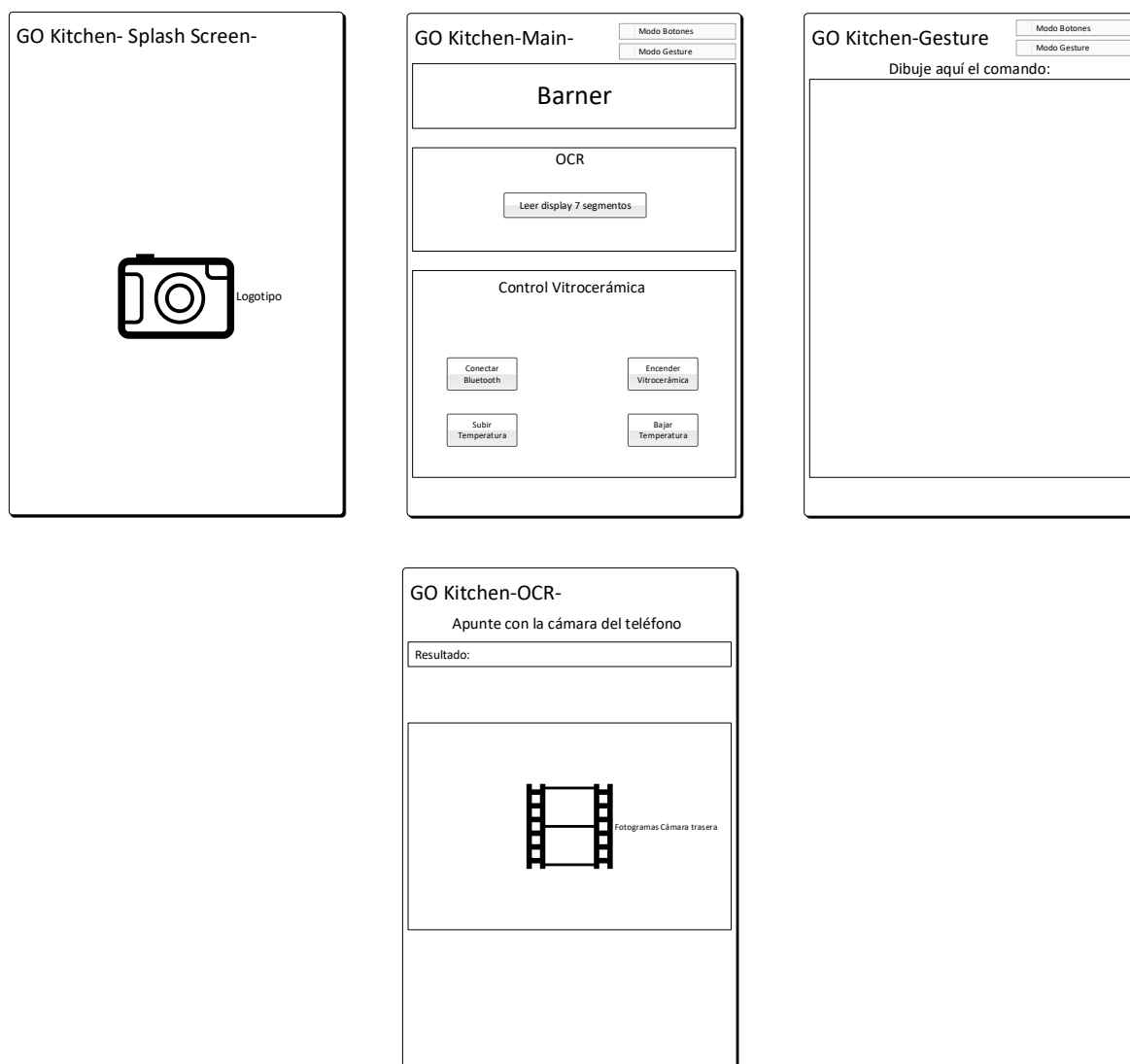


Ilustración 27: Wireframe final de la aplicación móvil.

4.5 Arquitectura de la unidad de control inteligente

La unidad de control inteligente es la parte perteneciente al segundo subsistema encargada de recibir las órdenes por parte de la aplicación Android y ejecutar una acción en la placa vitrocerámica en base al comando recibido.

En nuestro caso va a ser una raspberry Pi modelo 3 B. Viene equipada con muchas funcionalidades entre las que destacamos bluetooth, wifi y puertos GPIO que pueden ser configurados como entradas o salidas. Estos puertos estarán configurados de tal manera permitan enviar las señales necesarias para activar o desactivar funcionalidades en base a los comandos enviados.



Ilustración 28: Raspberry Pi 3 Modelo B

Esta placa electrónica es capaz de ejecutar un sistema Linux. Dentro del este sistema operativo ejecutaremos un script en Python que se puede consultar en el capítulo 6.4.3 y que nos va a permitir conectarnos y recoger los datos del aplicativo móvil. A su vez contará con la programación necesaria para mandar la señal necesaria a partir de los GPIO para activar el actuador correspondiente en la placa vitrocerámica.

4.5.1 Protocolo de comunicación, capa física y de transporte: Bluetooth, L2CAP

La capa de transporte de datos es L2CAP, (**Protocolo de control y adaptación del enlace lógico**). Es utilizado por la pila de protocolos de Bluetooth para pasar paquetes con y sin orientación a conexión a sus capas superiores. L2CAP se ocupa de:

- Segmentación y reensamblado de paquetes. Acepta paquetes de hasta 64KB de sus capas superior.
- Multiplexación de varias fuentes de paquetes, comprobando el protocolo de las capas superiores para así adaptarlo antes del reensamblaje.
- Proporcionar una buena gestión para la transmisión unidireccional a otros dispositivos bluetooth.
- Gestión de la calidad de servicio (QoS), del inglés Quality of service; para los protocolos de las capas superiores. En esta fase negocia el tamaño máximo del campo de datos de las tramas. Con ello, evita que algún dispositivo envíe paquetes tan grandes que puedan desbordar al receptor.

Por último, describir **bluetooth** como una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. Utilizaremos la versión 4.0 activa desde el año 2010 de este protocolo por su gran ancho de banda y mejoras en el consumo.

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores).

4.5.2 Protocolo de comunicación, capa de aplicación: *Binary Protocol*.

El protocolo usado sobre bluetooth para comunicarnos con la placa será un el protocolo Binary. Este protocolo como tal no es un estándar. Ha sido ideado para esta tesis establecer una comunicación segura y fiable entre el dispositivo móvil y el aplicativo bluetooth.

Cada trama del protocolo Binary consiste básicamente en la siguiente estructura:

CTX	Tipo de mensaje	Longitud	DATA	CRC	FTX
1 byte	1 byte	1 byte	1-255 Bytes	1 byte	1 byte
0x0A	0x00/0x01	0x00-0xFF	-	(^nbytes)- CTX-FTX	0x0B

Tabla 61: Binary Protocol.

- **[CTX] Comienzo de la trama:** Byte de comienzo de la trama, por convenio se ha elegido el byte 0x0A.
- **Tipo de mensaje:**
 - 0x00: mensaje de Tx desde Smartphone a Raspberry.
 - 0x01: mensaje de Rx desde Raspberry a Smartphone.
- **Longitud:** Indica la longitud de comando íntegramente comprendido en el campo DATA. El byte puede ser entre 0x00 y 0xFF.
- **DATA:** Comando a enviar en formato ASCII.
- **CRC:** Es la operación Xor de todos los bytes de la trama con excepción de los bytes CTX y FTX.
- **[FTX] Fin de la trama:** Byte de la trama, por convenio se ha elegido 0x0B.

Este protocolo contiene control de errores y puede ser ampliado fácilmente implementando nuevos comandos.

NOTA: para este TFM, este protocolo es opcional a implementar y sirve como complemento de fiabilidad y medio de comprobación de los mensaje y operación de envío. De hecho, es posible enviar los comandos son encapsular tal y como se muestra en el capítulo: 6.3.3.1

4.5.2.1 Lista de comandos implementados.

Esta es la lista de comandos implementados:

Comando	Enviado por	Explicación
STATUS	Smartphone > Raspberry Pi	Inicio de la comunicación entre Dispositivo móvil y placa Raspeberry Pi.
ON	Smartphone > Raspberry Pi	Encendido de placa vitrocerámica.
OFF	Smartphone > Raspberry Pi	Apagado de placa vitrocerámica.
PWUP	Smartphone > Raspberry Pi	Incremento de potencia.
PWDOWN	Smartphone > Raspberry Pi	Decremento de potencia.
OK	Raspberry Pi > Smartphone	Comando ejecutado correctamente
NOK	Raspberry Pi > Smartphone	Comando ejecutado con errores.

Tabla 62: Lista de Comandos disponibles.

CAPÍTULO 5: JUSTIFICACIÓN DE TECNOLOGÍAS USADAS Y PREPARACIÓN DEL ENTORNO DE DESARROLLO

5.1 Introducción

En esta parte del documento vamos a realizar una breve descripción de los elementos de hardware y software que se han empleado para la realización del proyecto. Posteriormente se hará una breve justificación acerca de la razón de uso de dichas tecnologías.

Para comenzar vamos a resumir todas las tecnologías resumidas en la siguiente tabla:

Elemento	Tipo	Descripción	Versión / Modelo
Hardware	Ordenador Portátil	Intel Pentium I7 4702MQ, 8GB RAM	Lenovo Thinkpad E540
“	Microcontrolador	A 1.2GHz 64-bit quad-core ARMv8 CPU, 1GB RAM, 40 GPIO pins	Raspberry Pi 3 Model B
“	Placa vitrocerámica	Inducción portable sin instalación	Bomann EKI 5026 CB
“	Smartphone	Qualcomm® Snapdragon™ 810, 4GB Ram, Cámara Frontal 5 MPx / Trasera 13 MPx	OnePlus 2
Software	Sistema Operativo Smartphone	Android	6.0.1
“	Sistema Operativo Microcontrolador	Linux Raspbian	Jessie 4.4
“	Lenguaje de Programación	Java	8.0
“	Lenguaje de programación	Python	3.4
“	Lenguaje de programación	C++	2012
“	IDE de desarrollo	Eclipse	Neon.2 Release (4.6.2)
“	Tratamiento de imágenes	OpenCv for Android	3.2
“	Motor OCR	Tesseract	3.4
“	Sistema Operativo para desarrollo	Linux Ubuntu	16.04

Tabla 63: Software y Hardware necesario para realizar el desarrollo

Para la elaboración de la documentación y redacción de esta memoria se usó:

- Microsoft Office ® Word 2016
- REM 1.2.2. Gestor de requisitos desarrollado por Amador Durán, para las fases de elicitación y análisis de requisitos.
- GIMP 2.8.18 para la elaboración de imágenes y retoque de fotografías e ilustraciones de este documento.
- Adobe Photoshop para la elaboración del logo.
- Microsoft Office ® Visio 2016, para la elaboración de los diagramas UML, diagramas de casos de uso y operaciones del sistema.
- Microsoft Office ® Powerpoint 2016, para la elaboración de la presentación que se expondrá en la defensa.
- Para el control de versiones se ha utilizado GitHub cuya dirección del repositorio donde se encuentra el código fuente es esta:
<https://github.com/arundil/OCR7Segments> . Para tratar con GitHub se usó el plugin de Git que viene por defecto en eclipse.

5.2 Razones por las que se ha optado por Android como sistema operativo para el desarrollo.

Android [16] es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS.

Con cerca de 297 millones de unidades a finales de 2016 según [18], Android es el sistema operativo orientado a dispositivos inteligentes más usado del mundo, seguido de lejos por IOS de Apple con algo más de 44 millones de unidades.

Aquí se aprecia la evolución de cuota de mercado de Android desde su aparición en 2008:

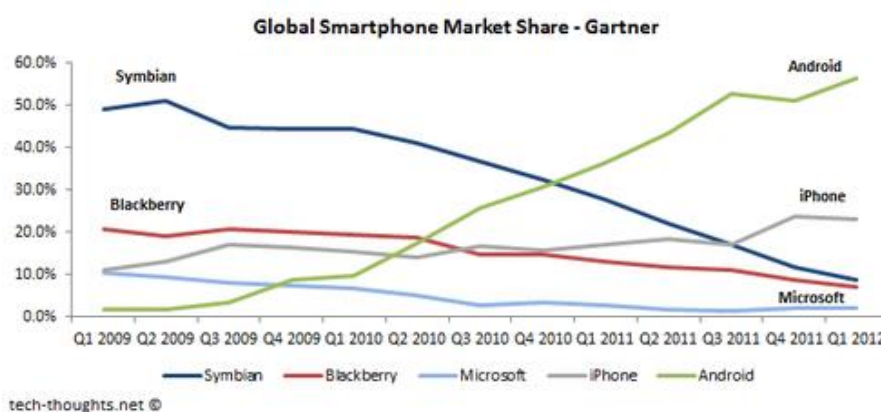


Ilustración 29: Evolución de la cuota de mercado de los sistemas operativos para teléfonos inteligentes, reproducido de [19]

Por esta razón entre otras se ha elegido para desarrollar este proyecto el sistema operativo Android.

5.3 Razones por las que se optado por Raspberry Pi como microcontrador

Raspberry Pi [20] es un computador de placa reducida, computador de placa única o computador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El software que utiliza es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite otros sistemas operativos, incluido una versión de Windows 10. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo B, y el 4 de febrero de 2013 del modelo A.

Raspberry pi se ha elegido especialmente por su facilidad de uso, y por el bajo coste de las placas electrónicas. Existen en a través de la red muchos tutoriales que facilitan su familiaridad y uso. En este proyecto se estaba buscando una placa inteligente que también disponga de bluetooth. El último modelo de Raspberry Pi nos ofrece todo el conjunto completo con Bluetooth y 40 GPIOs para poder configurarlos y conectarlos de forma sencilla a la placa vitrocerámica.

Es cierto que Raspberry Pi, a pesar que ejecuta un sistema operativo de software libre, la parte hardware es propietaria de Raspberry Pi Foundation, no siendo posible su uso comercial sin consentimiento de dicha organización. Si en un futuro se llegase a completar el desarrollo completo de este proyecto y por un casual llegase a comercializarse, se debería tener en cuenta este punto y quizá sustituir Raspberry Pi por otra placa electrónica que si sea de uso libre. Por otro lado, es cierto que para los fines de este trabajo fin de master Raspberry Pi cumple perfectamente las características que y funciones que necesitamos para la comunicación entre la aplicación y la placa vitrocerámica justificado así su elección.

5.3.1 Elección de Raspbian como sistema operativo de Raspberry Pi.

Como bien sabemos del apartado anterior, Raspberry pi permite ejecutar un sistema operativo bajo un núcleo ARM. Entre los Sistemas operativos que siguen directivas ARM se ha elegido uno familiar a su uso y basado en Debian como Raspbian.

Raspbian [21] es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Wheezy (Debian 7.0) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática. El lanzamiento inicial fue en junio de 2012.

La distribución usa **LXDE** [22] como escritorio y Midori como navegador web. Además, contiene herramientas de desarrollo como IDLE para el lenguaje de programación **Python** o Scratch, y diferentes ejemplos de juegos usando los módulos Pygame.

Una de las razones principales de su elección es por la capacidad de ejecutar scripts de Python a través del terminal. Esto nos permite correr servicios como el de Bluetooth y también activar los GPIO de manera sencilla.

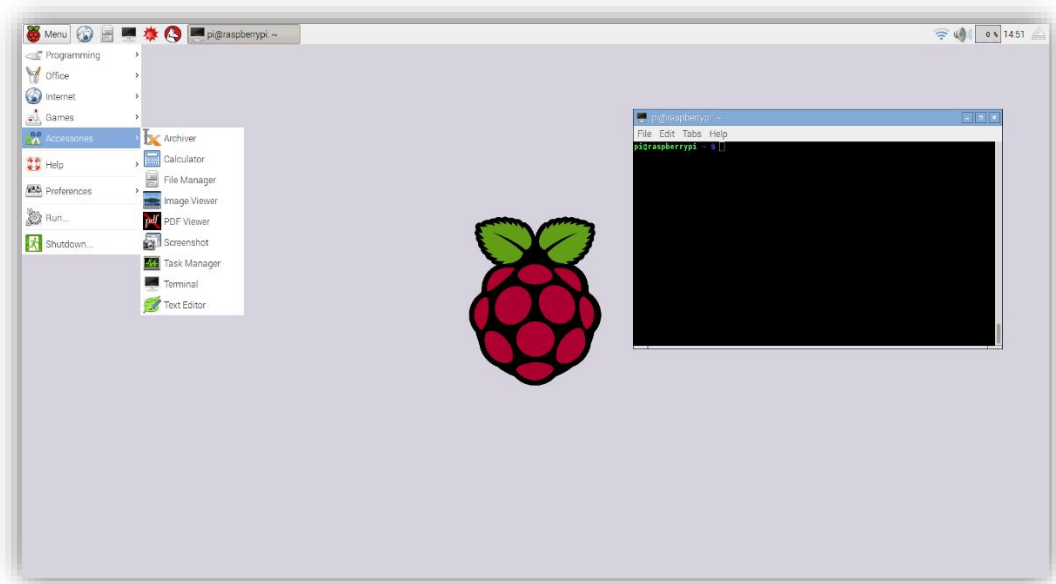


Ilustración 30: Captura de pantalla del sistema operativo Raspbian.

5.4 Elección de Tesseract como motor OCR.

Tesseract es actualmente uno de los mejores motores OCR más exactos de los que existen. Tesseract fue desarrollado en origen por Hewlett Packard entre 1985 y 1995. Después de 10 años sin desarrollo, su código fue liberado como código abierto. Esta ventaja la aprovechó Google, que tomó el desarrollo del mismo a partir del año 2006 hasta la fecha de redacción de esta memoria. Actualmente es el motor OCR que se usa en aplicaciones como "Google Translator". Permite a tiempo real traducir textos con la cámara del teléfono justo como se muestra en esta imagen:

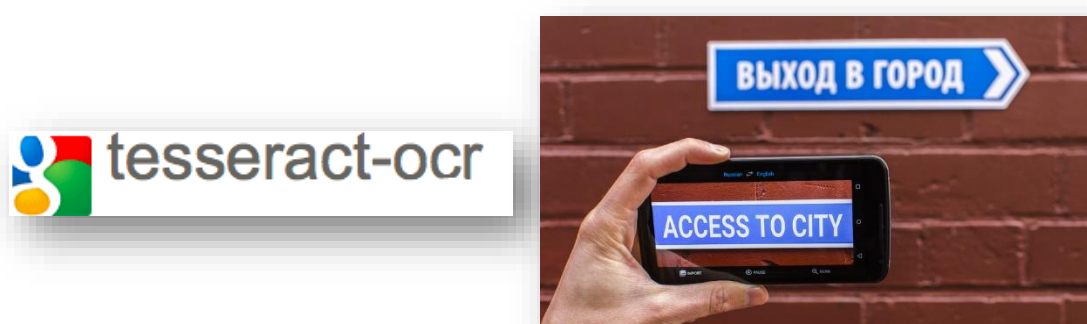


Ilustración 31: Logotipo y usos de Tesseract sobre Google Translator, reproducido de [35]

Tesseract es uno de los pocos motores OCR que admite entrenamiento para aprender a reconocer otros lenguajes o símbolos que no se encuentran en la lista de lenguajes disponibles. Esta característica es fundamental en este proyecto, ya que necesitamos reconocer símbolos representados por displays 7 segmentos. Esta condición ha sido determinante a la hora de seleccionar este motor OCR.

Tesseract no es un motor fácil de usar ni de integrar, pero su potencia, su capacidad poder entrenarlo con un lenguaje nuevo y su naturaleza de software libre lo han hecho propicio para este proyecto.

Apuntar de nuevo que para usarlo bajo la plataforma Android, se ha tenido que usar un "wrapper" que se describe en la sección: 4.4.1.2

5.5 Razones por las que se usó OpenCV como capa de tratamiento de imágenes.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCv es además una librería multiplataforma, capaz de ejecutarse sobre Android con facilidad. Es la librería que presenta los resultados más eficientes usando primitivas integradas de rendimiento para procesadores Intel.

Para este proyecto, era primordial encontrar un sistema que nos permitiera recortar un área de interés de cada fotograma capturado de la cámara. Para ello se necesita un software que detectase cuadrados de forma rápida ya que se requieren de varias capturas para reconocer el dato de forma correcta.

OpenCV además es open source y su distribución y uso no conllevan infracción de copyright.

5.6 Razones por la que se ha usado Python en Raspberry Pi para el servicio bluetooth.

Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Según el ranking Tiobe, Python ocupa el 5º lugar en los lenguajes más utilizados del mundo. Su evolución ha sido positiva en los últimos años debido a la aparición de buenas librerías que permiten realizar muchas funcionalidades con código compacto.

Feb 2017	Feb 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.676%	-4.47%
2	2		C	8.445%	-7.15%
3	3		C++	5.429%	-1.48%
4	4		C#	4.902%	+0.50%
5	5		Python	4.043%	-0.14%
6	6		PHP	3.072%	+0.30%

Python ha demostrado un repunte en los últimos años debido también a la aparición de microcontroladores que aceptan cada vez más este tipo de lenguaje. Además, cabe resaltar que es multiplataforma, con lo que tu código es fácilmente exportable y ejecutable en diferentes máquinas.

En este caso, Python ha sido utilizado como gestor de servicios bluetooth que estará a la escucha de comandos que lleguen por parte de la aplicación móvil. En el próximo capítulo se explicará el uso del script creado para tal efecto.

5.7 Uso de Java para la interfaz de usuario de la aplicación Android.

Para interfaces de usuario existen varios frameworks que se puede usar en la actualidad. Al principio estuve barajando el uso de Xamarin [23]. Xamarin es una plataforma que permite, bajo código en C#, escribir código para aplicaciones que se ejecuten tanto para Android como para IOS. La idea original era tener la aplicación funcionando en ambas plataformas, con lo que Xamarin se ajustaba perfectamente a las necesidades. El problema llegó justo a la hora de la intergración con Tesseract. No existían “wrappers” disponibles que permitiesen la rápida integración de tesseract en Xamarin. Tras buscar mucha información, al final decidí posponer el desarrollo para plataforma IOS y migré la aplicación completa a Java, dónde sí que existía Tess-two [13].

Java también es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o “write once, run anywhere”), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Actualmente es el lenguaje de programación más usado del mundo según el ranking Tiobe [24].

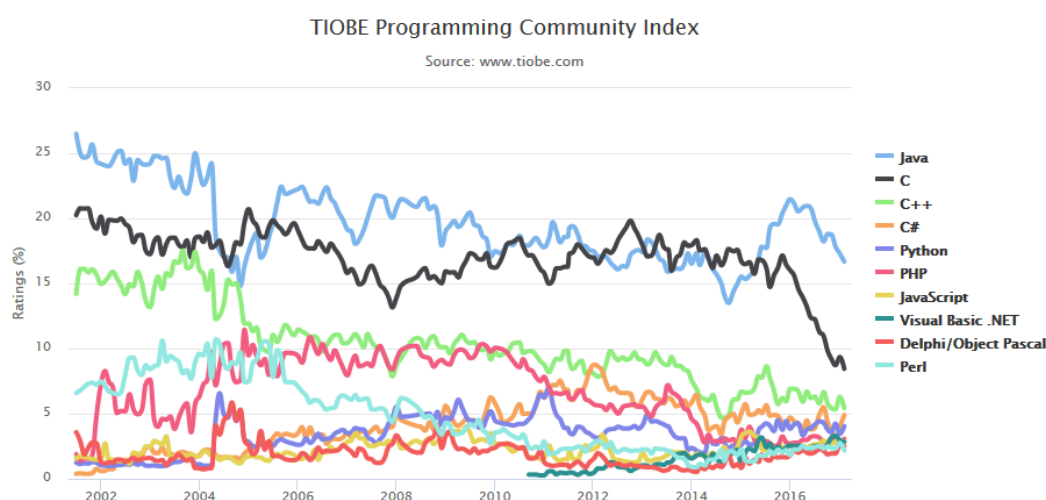


Ilustración 32: Evolución del uso de los lenguajes de programación más populares, reproducido de [24]

5.8 Eclipse como IDE para Android

Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc...

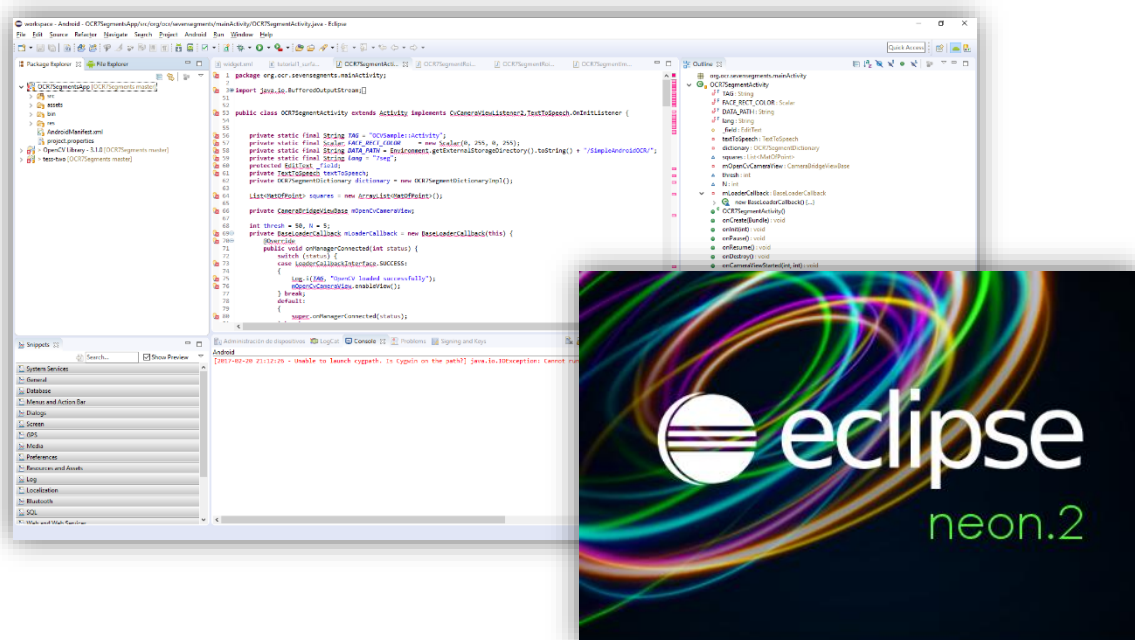


Ilustración 33: Eclipse

Me decidí a usar Eclipse para Android descartando Android Studio de google por su incompatibilidad con Tesseract y OpenCV. Ambos necesitan un compilador de c++ generar el ejecutable. Eclipse fue el único IDE que incorporaba todas las funcionalidades necesarias para poder compilar Java y C++ a la vez. También posee una potente herramienta para diseñar interfaces para Android.

5.8.1 Configuración de eclipse para trabajar con Android, Tesseract y OpenCV.

Para trabajar con eclipse primero hemos de descargarlo. Existe una versión para desarrolladores de Android en la siguiente URL:

<http://www.eclipse.org/downloads/packages/eclipse-android-developers/neonm6>

Una vez descargado, descomprimiremos la carpeta en el lugar deseado y basta con pulsar sobre el ejecutable que se encuentra dentro de la carpeta descargada para ponerlo en marcha.

Es necesario tener Java 1.7 JRE instalado o una versión superior. En caso de no tenerlo instalado debemos descargarlo e instalarlo desde esta URL:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Una vez instalado Eclipse para Android, nos pedirá configurar un SDK para Android. En este caso nos saldrá el siguiente asistente:

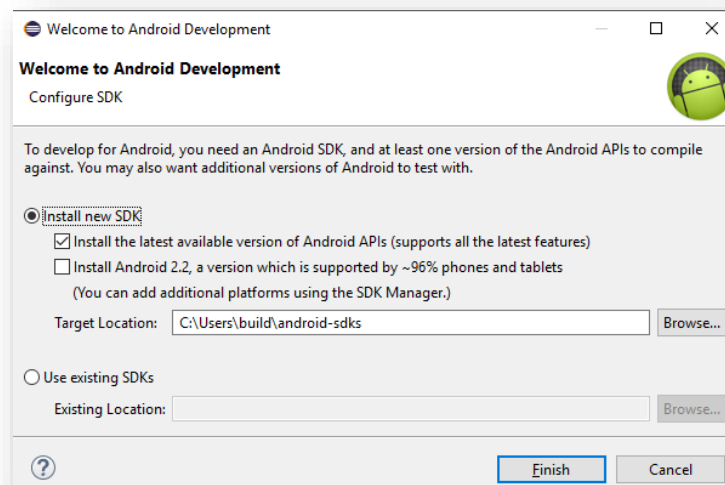


Ilustración 34: Asistente de configuración del SDK.

Instalamos la última versión del SDK de Android. Con ella nos garantizamos que la aplicación funcione en dispositivos de última generación. Aceptamos las licencias e instalamos el SDK. Es cierto la aplicación ha sido desarrollada para un SDK 21 de Android (equivalente a Android 4.1). El SDK mínimo para funcionar es el SDK 8 de Android.

Nos bajamos el proyecto de github. Podemos hacer un clone del repositorio o ir directamente a:

<https://github.com/arundil/OCR7Segments/tree/development-AndroidApp>

Y descargarse el código fuente en un fichero Zip.

Acto seguido importamos los proyectos a Eclipse. Eclipse nos mostrará una serie de errores que debemos solventar, aunque para eso vamos por partes:

5.8.1.1 *Compilar Tess-Two.*

Para poder compilar Tess-Two hace falta usar un compilador de C o C++ nativo para Android. Es por ello que usaremos Android NDK [25]. Android NDK es un conjunto de herramientas que te permite implementar partes de una app usando lenguajes de código nativo como C y C++.

Una vez descargado Android NDK, es necesario volver a Eclipse e ir a Preferences → Android → NDK y configurar el directorio donde hayamos descargado la API de Android NDK. El directorio debe ser algo similar a “\android-ndk-r14b\build”

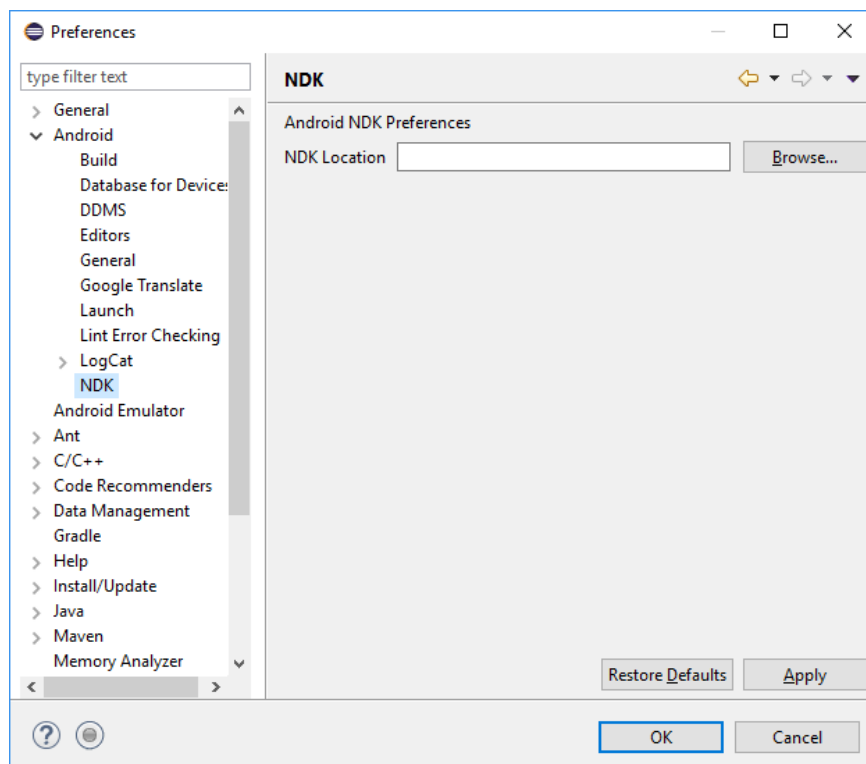


Ilustración 35: Configuración de NDK de Android en Eclipse.

Acto seguido, hacemos clic derecho sobre el proyecto Tess-Two y accedemos a propiedades → Builders y otorgamos un nombre al constructor como “NDK tess” y asociamos la ruta donde tenemos instalado el repositorio NDK. Como “workspace” escribimos: “\${workspace_loc:/tess-two}”

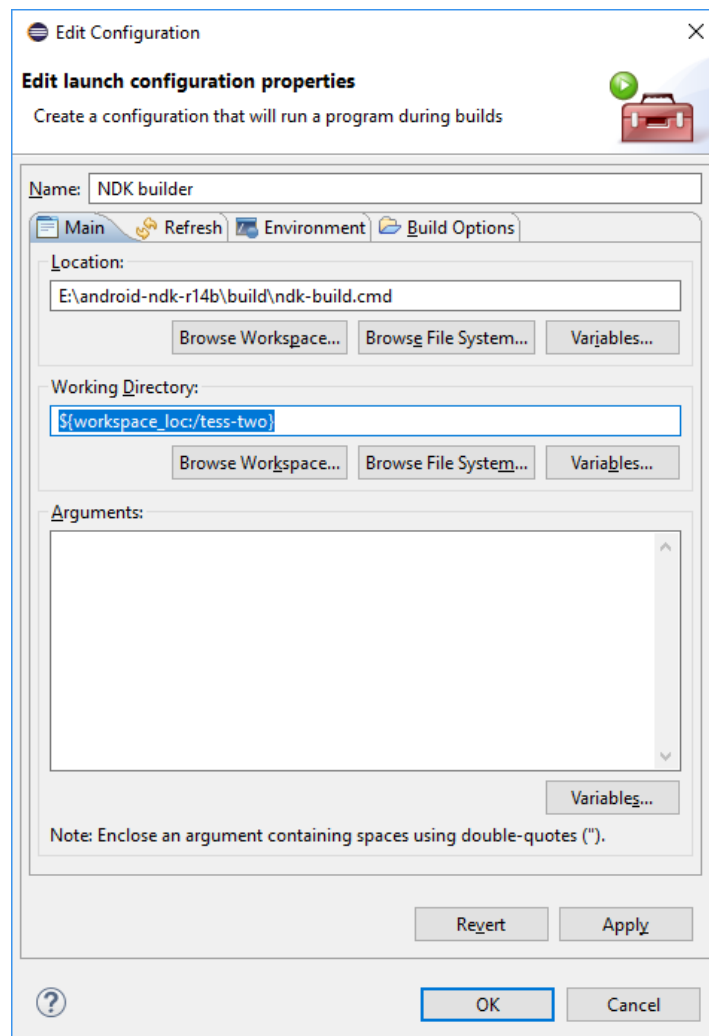


Ilustración 36: Configuración de compilación del código de Tess Two usando NDK.

Refresca el proyecto de tess-two y espera unos 10~12 minutos a que termine de compilarse (esto depende la velocidad de tu ordenador).

Nota importante: En caso de usar **NDK versión 14** o superior, debido a que ahora son necesarias incluir todas las dependencias para compilar el proyecto, es necesario añadir la siguiente directiva al fichero `Application.mk`:

```
APP_ALLOW_MISSING_DEPS := true
```

Más info en esta fuente [37]

Esto es solo necesario si al compilar obtenemos el siguiente error:

```
"(..)Module pngt depends on undefined modules(..)"
```

Después de que acabe, añadimos el proyecto como una librería a nuestro proyecto de "Go Kitchen" ambos proyectos como librerías. Para ello hacemos clic derecho sobre nuestro proyecto y vamos a propiedades. Allí seleccionamos el menú Android y seleccionamos la API (recomendada 7.1.1) y abajo seleccionamos los dos proyectos como librerías, tanto Tess Two como Opencv.

Con esto tendremos listo el entorno para desarrollar sobre la aplicación. También es posible subir la aplicación a un dispositivo Android tan solo ejecutándola con el botón correspondiente en Eclipse y conectado un dispositivo vía USB al ordenador donde estemos desarrollando.

CAPÍTULO 6: IMPLEMENTACIÓN

6.1 Introducción

En este capítulo se explica la implementación del proyecto. Está dedicado plenamente a la fase de desarrollo. Hacer hincapié que este capítulo se ha redactado al finalizar dicha fase, previo a la realización de pruebas del sistema.

Para descargar el código fuente es recomendable recurrir al repositorio de Github donde existe también documentación (en inglés) del aplicativo.

La parte de código que se va a explicar pertenece a la parte implementada en la aplicación móvil, así como el script que contiene la parte de Bluetooth dentro de Raspberry Pi.

En esta parte introductoria se incluye el logotipo de la aplicación diseñado a tal efecto para que sirva de imagen corporativa del aplicativo.



Ilustración 37: Logotipo de la aplicación

Finalmente, el nombre elegido del aplicativo ha sido “Go Kitchen”. Tiene un toque comercial para que sea fácil de recordar por el público en general a la hora de buscar la aplicación en Google Play.

6.2 Instalación de la aplicación en un dispositivo Android.

Esta aplicación es posible instalarla en cualquier dispositivo android con una versión superior a la API 8 (Android 2.1) usando Google Play, o el repositorio de código fuente github:

6.2.1 Modo sencillo a través de Google Play.

Google Play es actualmente la plataforma de aplicaciones móviles.

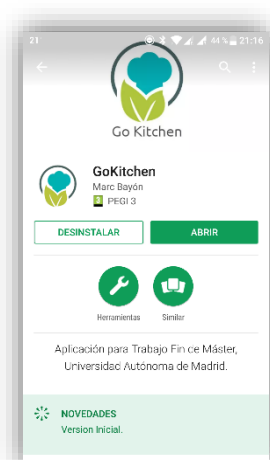


Ilustración 38: Go Kitchen en Google Play.

Una vez instalada la aplicación, si queremos usar la parte de OCR, será necesario instalar las librerías de OpenCV:

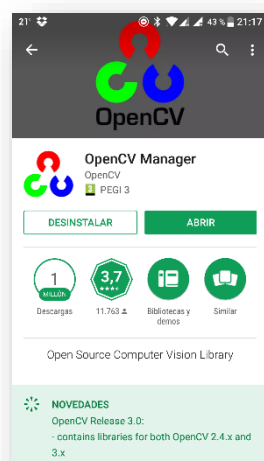


Ilustración 39: OpenCv en Google Play

Una vez realizado la aplicación debe iniciarse con normalidad.

6.2.2 Modo desarrollador a través de Eclipse.

Para instalar la aplicación con Eclipse, es necesario primero clonar el repositorio de gitHub. Aquí se explica cómo hacerlo con un sistema operativo Linux. Para Windows será necesario descargarse git y también los drivers del dispositivo móvil android a utilizar.

Abrir un terminal bash y escribir el siguiente comando:

```
:~$ git clone https://github.com/arundil/OCR7Segments.git
```

Una vez clonado el repositorio, hay que cambiar de branch, ya que el código fuente de la aplicación no se encuentra en el Master branch. Para ello usar el siguiente comando:

```
:~$ git branch development-AndroidApp
```

Una vez cambiado el branch, abrimos Eclipse e importamos el proyecto. Hay que tener en cuenta que antes de importar el proyecto, eclipse ha debido ser configurado tal y como se explica en el apartado 5.8.1 .

Importamos los siguientes proyectos:

- GoKitchen
- Opencv Library
- Tess-two

Una vez importados, tan solo hay que conectar por USB el Smartphone en modo desarrollador. Una vez lo tenemos conectado, tan solo hay que depurar el proyecto y eclipse nos instalará el fichero APK necesario para ejecutar la aplicación.

6.3 Vistazo a la implementación de la aplicación Android.

En este apartado vamos a explicar de forma resumida como se ha implementado la aplicación

Todo el código de esta aplicación se encuentra a libre disposición de aquel que lo necesite en GitHub, en la siguiente dirección web: <https://github.com/arundil/OCR7Segments>

6.3.1 Estructura de clases de la aplicación principal.

Cada aplicación Android divide cada pantalla en clases denominadas "Activity" en las que se programa la lógica de negocio pertinente a los objetivos definidos en el apartado de análisis de requisitos (véase capítulo 3.2). En este punto analizaremos, sin entrar en detalles de implementación, el funcionamiento de las principales clases principales de la aplicación.

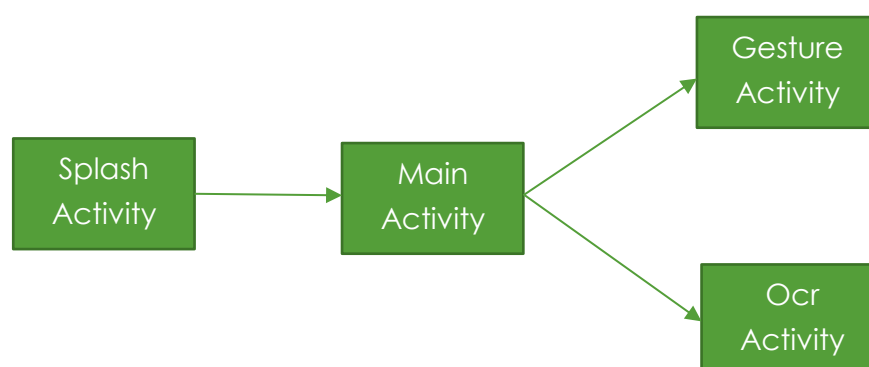


Ilustración 40: Esquema de navegación de la aplicación móvil

6.3.2 Splash activity

Esta actividad es la encargada de mostrar una pantalla de espera mientras se cargan todas las librerías necesarias para arrancar la aplicación.

Esta actividad se ejecuta de forma automática y no requiere interacción por parte del usuario.

Aquí se muestra la parte de código que se encarga de esta funcionalidad:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Set portrait orientation
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_
    PORTRAIT);
    // Hide title bar
    requestWindowFeature(Window.FEATURE_NO_TITLE);

    setContentView(R.layout.activity_splash);

    textToSpeech = new TextToSpeech( this, (OnInitListener) this
);
    textToSpeech.setLanguage( new Locale( "spa", "ESP" ) );

    TimerTask task = new TimerTask() {
        @Override
        public void run() {

            // Start the next activity
            Intent mainIntent = new Intent().setClass(
                SplashActivity.this, MainActivity.class);
            startActivity(mainIntent);

            // Close the activity so the user won't able to go
            // back this
            // activity pressing Back button
            finish();
        }
    };
};
```

Y este es el resultado final de la pantalla:

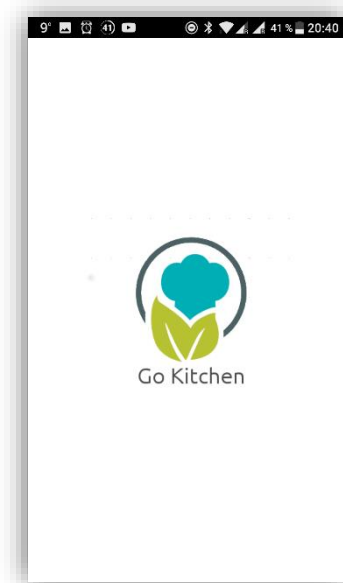


Ilustración 41: Splash Screen

6.3.3 Main Activity

Esta es la pantalla principal de la aplicación desde donde se puede acceder a todos los subsistemas implementados y descritos en la arquitectura de la aplicación (véase capítulo 4.2). En esta clase se implementan varias funcionalidades clave:

- Conexión Bluetooth con Raspberry Pi.
- Acceso al modo accesibilidad y OCR
- Comandos para manejo de la placa.

Pasamos a describir cada funcionalidad por separado.

6.3.3.1 Implementación del cliente bluetooth.

La principal funcionalidad de esta pantalla es la interoperabilidad, comunicación y manejo de datos para comunicarse con el servidor bluetooth (cuya implementación se explica más adelante).

Para implementar la funcionalidad de bluetooth me he basado en la documentación ofrecida por Google Developers referenciada aquí: [26]

Los pasos seguidos para la implementación son los siguientes:

- 1) Obtener el manejador del dispositivo bluetooth

```
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  
  
arrayLog = getResources().getStringArray(R.array.textStatus);  
  
if (mBluetoothAdapter == null) {  
    // Device does not support Bluetooth  
    AlertDialog alertDialog = new  
AlertDialog.Builder(MainActivity.this).create();  
    alertDialog.setTitle("Oh Oh!");  
    alertDialog.setMessage("Your device does not support  
bluetooth");  
    alertDialog.setButton(AlertDialog.BUTTON_NEUTRAL, "OK",  
        new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int which)  
            {  
                dialog.dismiss();  
            }  
        });  
    alertDialog.show();  
}
```

- 2) Tratar de obtener el enlace con el dispositivo externo, chequear el estado.

```
private Boolean checkBTState() {
    // Check for Bluetooth support and then check to make sure it
    is turned on

    if (mBluetoothAdapter.isEnabled()) {
        Log.d(TAG, "...Bluetooth is enabled...");
        return true;
    } else {
        //Prompt user to turn on Bluetooth
        Intent enableBtIntent = new
Intent(mBluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
        return true;
    }
}
```

- 3) Conectarse con el dispositivo externo. Primero es necesario emparejarse los dispositivos y abrir un canal de comunicación. De eso se encarga este fragmento de código.

```
private Boolean connectBT() {

    Log.d(TAG, "...In onResume - Attempting client connect...");

    // Set up a pointer to the remote node using it's address.
    Set<BluetoothDevice> pairedDevices =
mBluetoothAdapter.getBondedDevices();
    String deviceHardwareAddress = "";
    if (pairedDevices.size() > 0) {
        // There are paired devices. Get the name and address of
each paired device.
        for (BluetoothDevice device : pairedDevices) {
            if (device.getName().equals("raspberrypi")) {
                deviceHardwareAddress = device.getAddress(); //
MAC address
                break;
            }
        }
    }
    loginfo.setText(arrayLog[1]);
    BluetoothDevice device = null;
    if (!(deviceHardwareAddress == ""))
        device =
mBluetoothAdapter.getRemoteDevice(deviceHardwareAddress);
    else
        device = mBluetoothAdapter.getRemoteDevice(address);
}
```

```

        // Two things are needed to make a connection:
        //   A MAC address, which we got above.
        //   A Service ID or UUID. In this case we are using the
        //   UUID for SPP.
        try {
            mBluetoothSocket =
device.createRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) {

            errorExit("Fatal Error", "In onResume() and socket create
failed: " + e.getMessage() + ".");
            return false;
        }

        // Discovery is resource intensive. Make sure it isn't going
on

        // when you attempt to connect and pass your message.
        mBluetoothAdapter.cancelDiscovery();

        // Establish the connection. This will block until it
connects.
        Log.d(TAG, "...Connecting to Remote...");
        try {
            mBluetoothSocket.connect();
            Log.d(TAG, "...Connection established and data link
opened...");
        } catch (IOException e) {
            try {
                mBluetoothSocket.close();
            } catch (IOException e2) {
                errorExit("Fatal Error", "In onResume() and unable to
close socket during connection failure" + e2.getMessage() + ".");
                return false;
            }
            return false;
        }

        // Create a data stream so we can talk to server.
        Log.d(TAG, "...Creating Socket...");

        try {
            outputStream = mBluetoothSocket.getOutputStream();
        } catch (IOException e) {
            errorExit("Fatal Error", "In onResume() and output stream
creation failed:" + e.getMessage() + ".");
            return false;
        }

        return true;
    }

```


4) Una vez establecida la conexión, nos queda el envío de datos:

```
private Boolean sendData(String message) {
    byte[] msgBuffer = message.getBytes();

    Log.d(TAG, "...Sending data: " + message + "...");

    try {
        outputStream.write(msgBuffer);
        return true;
    } catch (IOException e) {
        String msg = "In onResume() and an exception occurred
during write: " + e.getMessage();
        if (address.equals("00:00:00:00:00:00"))
            msg = msg + ".\n\nUpdate your server address from
00:00:00:00:00:00 to the correct address on line 37 in the java code";
        msg = msg + ".\n\nCheck that the SPP UUID: " +
MY_UUID.toString() + " exists on server.\n\n";

        errorExit("Fatal Error", msg);
        return false;
    }
}
```

Con esto tenemos todas las funciones de la capa física necesarias para comunicarnos con el dispositivo externo.

6.3.3.2 Implementación de los mensajes de comunicación entre aplicación móvil y dispositivo bluetooth.

Siguiendo la lista de comandos definidos en el apartado 4.5.2.1 a continuación se muestra la parte de código que corresponde al cliente bluetooth que desempeña la aplicación Android.

El protocolo a seguir está descrito también en la sección 4.5.2 conocido como binary protocol. Aquí se muestra un fragmento de código de la lógica para enviar el mensaje de cada botón de la aplicación. Todos los botones están programados de la misma manera con lo que simplemente se ha replicado la misma estructura por cada botón. Aun así, se le invita al lector si necesita de más información, recurra al repositorio de código fuente en GitHub.

He aquí la implementación de uno de los botones tipo Switch que contiene la aplicación.

```

final Switch OnOff = (Switch) findViewById(R.id.OnOff);
OnOff.setChecked(false);

OnOff.setOnCheckedChangeListener(new OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        if (mBluetoothAdapter != null &&
mBluetoothAdapter.isEnabled()) {

            if(ConnectBT.isChecked()) {

                if(isChecked){
                    sendData("ON");
                    Toast.makeText(getApplicationContext(),"ON",
Toast.LENGTH_SHORT).show();
                }
                else {
                    sendData("OFF");
                    Toast.makeText(getApplicationContext(),"OFF",
Toast.LENGTH_SHORT).show();
                }
            }
            else {
                Toast.makeText(getApplicationContext(),"Please
connect first with the Electronic HOB", Toast.LENGTH_SHORT).show();
            }
        }
        else {
            Toast.makeText(getApplicationContext(),"Error on
Bluetooth", Toast.LENGTH_SHORT).show();
        }
    }

});

```

6.3.3.3 Vistazo a la interfaz final de Main Activity.

El resultado final de esta pantalla es la siguiente vista donde, a pesar del alto contenido, se ha dejado lo más accesible y clara posible:



Ilustración 42: Pantalla Main Activity

6.3.4 Gesture Activity

En esta pantalla se ha habilitado para el modo con accesibilidad. En ella se pueden configurar las mismas funcionalidades que en la pantalla principal, pero usando gestos en vez de botones. Para ello antes de implementar esta clase, es necesario generar el fichero de gestos. Para ello es necesario descargar algún programa que nos genere el fichero de gestos. Para este trabajo fin de máster, se ha elegido la aplicación “Gesture Builder” [27].

Aquí se muestra una captura de la aplicación:



Ilustración 43: Aplicación Gesture builder

Esta aplicación nos genera un fichero de texto que podemos cargar desde la API de Android de Gesture Detector.

El fichero hay que ponerlo sobre la carpeta res/raw donde se podrá rescatar más tarde con la ingeniería inversa de Android.

Aquí se muestra la inicialización de los parámetros:

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_gesture);

    mLibrary = GestureLibraries.fromRawResource(this,
R.raw.gesture);
    if (!mLibrary.load()) {
        finish();
    }

    GestureOverlayView gestures = (GestureOverlayView)
findViewById(R.id.gestures);

    gestures.addOnGesturePerformedListener((OnGesturePerformedListener)
this);
}

```

El manejo de los gestos se hace a través del evento “onGesturePerformed” tal y como se muestra a continuación:

```

public void onGesturePerformed(GestureOverlayView overlay, Gesture
gesture) {

    ArrayList<Prediction> predictions =
mLibrary.recognize(gesture);
    if (predictions.size() > 0 && predictions.get(0).score > 1.0)
    {
        String result = predictions.get(0).name;

        if ("ocr".equalsIgnoreCase(result)) {
            Toast.makeText(this, "OCR Mode",
Toast.LENGTH_LONG).show();
            Intent ocrActivity = new Intent(GestureActivity.this,
OCRActivity.class);
            startActivity(ocrActivity);

        }else if ("connect".equalsIgnoreCase(result)) {
            Toast.makeText(this,
"Conectado", Toast.LENGTH_SHORT).show();
        } else if ("on".equalsIgnoreCase(result)) {
            Toast.makeText(this, "ON", Toast.LENGTH_LONG).show();
        } else if ("off".equalsIgnoreCase(result)) {
            Toast.makeText(this, "OFF", Toast.LENGTH_LONG).show();
        } else if ("subir temperatura".equalsIgnoreCase(result)) {
            Toast.makeText(this, "Subir Temperatura",
Toast.LENGTH_LONG).show();
        } else if ("bajar temperatura".equalsIgnoreCase(result)) {
            Toast.makeText(this, "Bajar
Temperatura", Toast.LENGTH_SHORT).show();
        }

    }

}

```

6.3.5 OCR Activity

Esta es la actividad más importante y motivo principal del estudio de esta tesis. En esta actividad se define la parte del motor OCR y el tratamiento de imágenes. En esta actividad se hace uso de:

- Tesseract
- OpevCv
- Tratamiento de imágenes.

Esta actividad depende de los proyectos de OpenCV y Tess two. Para la configuración en Eclipse de ambos proyectos, en necesario consultar el capítulo 5.8.1

6.3.5.1 Inicialización de la actividad

La inicialización se divide en varias partes que se enumeran a continuación:

- Inicialización de la lectura por voz

```
textToSpeech = new TextToSpeech( this, this );
textToSpeech.setLanguage( new Locale( "spa", "ESP" ) );
```

- Inicialización de hardware de la cámara

```
mOpenCvCameraView = (CameraBridgeViewBase)
findViewById(R.id.surface_view);
/*Set the resolution to the lowest supported*/

mOpenCvCameraView.setMaxFrameSize(640, 480);

mOpenCvCameraView.setVisibility(View.VISIBLE);

mOpenCvCameraView.setCvCameraViewListener(this);
/*Set the resolution to the lowest supported*/

mOpenCvCameraView.setMaxFrameSize(640, 480);

mOpenCvCameraView.setVisibility(View.VISIBLE);
```

- Carga del fichero de entrenamiento del motor OCR

```

        if (!(new File(DATA_PATH + "tessdata/" + lang +
".traineddata")).exists()) {
            try {

                AssetManager assetManager = getAssets();
                InputStream in = assetManager.open("tessdata/" + lang
+ ".traineddata");
                //GZIPInputStream gin = new GZIPInputStream(in);
                OutputStream out = new FileOutputStream(DATA_PATH +
"tessdata/" + lang + ".traineddata");

                // Transfer bytes from in to out
                byte[] buf = new byte[1024];
                int len;
                //while ((len = gin.read(buff)) > 0) {
                while ((len = in.read(buf)) > 0) {
                    out.write(buf, 0, len);
                }
                in.close();
                //gin.close();
                out.close();

                Log.v(TAG, "Copied " + lang + " traineddata");
            } catch (IOException e) {
                Log.e(TAG, "Was unable to copy " + lang + "
traineddata " + e.toString());
            }
        }

        _field = (EditText) findViewById(R.id.field);

        mOpenCvCameraView.setCvCameraViewListener(this);

```

6.3.5.2 Análisis de cada fotograma capturado

Para buscar el texto se requiere de un análisis exhaustivo de cada fotograma capturado por el objetivo de la cámara de fotos.

Para ello, la actividad implementa un evento denominado "OnCameraFrame".

Por cada fotograma se aplicará el siguiente método:

- Encuentro del área de interés buscando áreas cuadradas en la imagen
Para ello se hace uso de la librería Opencv y de método "Imgproc.drawContours"

```
//Here we draw the squares (blue ones) for each ROI that we detect.  
Imgproc.drawContours(image, squares, -1, new Scalar(0,0,255));
```

- En caso de encontrarlos, analizar que el tamaño no supere los 380x95 y exceda de un tamaño de 172x44 pixeles para una resolución de 640 x 480.
Para ello, se selecciona una sub-imagen del fotograma donde se entiende que está el dato en cuestión.

En caso que la cámara del Smartphone no sea capaz de capturar una imagen con una resolución de 640 x480, se hace un factor de conversión para las medidas en función de la imagen de entrada:

```
for (MatOfPoint p :squares)  
{  
  
    Mat imageROI = image.submat(Imgproc.boundingRect(p));  
    //Calculate the resolution of every photogram  
    int resolution = image.width()/image.height();  
  
    Log.i(TAG, "W: "+image.size().width+" H: "+image.size().height);  
    Log.i(TAG, "RESOLUTION: "+resolution);  
    double reductionFactorW = 1;  
    double reductionFactorH = 1;  
  
    if (image.size().width ==352 && image.height() == 288)  
    {  
        reductionFactorW = 1.82;  
        reductionFactorH = 1.68;  
    }  
}
```



```

        //The square should not be large than 380x95 and
        bigger than 172x44 in resolution 640x480. if resolution is 352x288, I
        apply a reduction factor.
        //Remember than height and width is inverted in
        Portrait mode for the input image.

        if (((imageROI.height()<=(172/reductionFactorH) &&
        (imageROI.height()>=(44/reductionFactorH))))
            &&
        (((imageROI.width()<=(380/reductionFactorW) &&
        (imageROI.width()>=(95/reductionFactorW))))) {

            (...)
        }
        (...)
    }
}

```

- En caso de cumplirse este requisito, se recorta el área de interés y se procede al análisis, para ello se procede a preparar la imagen para pasarla al motor OCR. Este procedimiento se explica de forma teórica en el capítulo: 4.2.1, y se lleva a cabo en la función “prepare image4OCR”. Los pasos son los siguientes:

- Pasar de escala de Grises a Imagen Binaria usando el umbral de OTSU.

```

Imgproc.cvtColor(ret, ret, Imgproc.COLOR_RGBA2GRAY);
Imgproc.threshold(ret, ret, 0, 255, Imgproc.THRESH_BINARY_INV |
Imgproc.THRESH_OTSU); //Threshold put to 127 over 255

```

- Calcular la mediana de la imagen para aplicar un suavizado.

```

Imgproc.medianBlur(ret, ret, 5); //Smooth filter

```

- Enderezar la imagen en caso de ser necesario.

```

ret= OCRImage.deskew(ret);

```

- Quitar resto de líneas verticales y horizontales que se puedan encontrar.

```
int cols_to_remove = (int) (ret.cols()*0.05);
int rows_to_remove = (int) (ret.rows()*0.05);
Mat retfinal= ret.submat(rows_to_remove, ret.rows()-rows_to_remove,
cols_to_remove, ret.cols()-cols_to_remove);
```

- Erosionar la imagen con una matriz de píxeles 2x2.

```
Mat kernel = Mat.ones(new Size(2,2),CvType.CV_8U);
Imgproc.erode(retfinal, retfinal, kernel,new Point(),1);
```

- Al finalizar el tratamiento de la imagen se debe analizar si la imagen tiene un 70 % de píxeles blancos.

```
int WhitePixels = Core.countNonZero(imageROI_prepared);
float numOfPixels = imageROI_prepared.height() *
imageROI_prepared.width();
double avarageOfWhitePixels = (WhitePixels/numOfPixels)*100;

if (avarageOfWhitePixels > 70) {

    (...) (Ocr processing)
}
```

- En caso de superar esta ultimo filtro se pasa la imagen resultante al motor OCR con el fichero de entrenamiento previamente cargado.

```
TessBaseAPI baseApi = new TessBaseAPI();
baseApi.setDebug(true);
baseApi.init(DATA_PATH, lang);
baseApi.setImage(bitmap);
//Printing the result in the GUI
final String recognizedText = baseApi.getUTF8Text();
baseApi.end();
```

- Finalmente, en caso de obtener resultado positivo, este se compara con el resultado de valores posibles.

```
dictionary.UpdateElement(recognizedText, 1);
```

- Si existe relación y se han analizado al menos 5 fotogramas de forma correcta, entonces se confirma la lectura correcta del número a través del altavoz del teléfono. En cualquier caso, el resultado siempre se muestra por la interfaz de usuario en un hilo paralelo de ejecución tal y como se muestra en el siguiente fragmento de código:

```

if ( recognizedText.length() != 0 ) {

    this.runOnUiThread(new Runnable() {

        public void run() { //GUI thread

            _field.setText(_field.getText().toString().length()
                == 0 ? recognizedText : recognizedText);

            _field.setSelection(_field.getText().toString().length());
            textToSpeech.setLanguage( new Locale( "esp", "ESP" ) );
            String value = dictionary.evaluateDictionary();
            if (!value.equals("")){
                speak(value);
                dictionary.restartDictionary();
            }

        }

    });
}

```

6.3.5.3 Detección de cuadrados por aproximación de polinomios y detección de ángulo.

Para complementar el apartado anterior, aquí se explica el método utilizado para la detección del área de interés (cuadrada) a partir de aproximación de rectángulos por detección de puntos a partir de OpenCv.

- Primeramente, se recogen 2 imágenes con la misma información, pero una de ellas con resolución $N/2$ respecto a la original. A partir de aquí a la original se le aplica un filtro piramidal de reducción de escala, y a la pequeña se le aplica un filtro piramidal de ampliación de escala. Con este procedimiento conseguimos reducir el ruido de la imagen. Más información en esta fuente: [28]

```

//Image that we need

Mat smallerImg=new Mat(new Size(inputImage.width()/2,
inputImage.height()/2),inputImage.type());

Mat gray = new Mat(inputImage.size(),inputImage.type());

Mat gray0 = new Mat(inputImage.size(),CvType.CV_8U);

//down-scale and upscale the image to filter out the noise

Imgproc.pyrDown(inputImage, smallerImg,smallerImg.size());

Imgproc.pyrUp(smallerImg, inputImage,inputImage.size());

```

- Luego, se extrae uno de los tres canales de la imagen (a saber, entre Red, Green o Blue). Para reducir procesamiento, en la aplicación solo extraemos la imagen del canal Red. Para obtener mejores resultados es aconsejable buscar en todos los canales de la imagen.

```
extractChannel(inputImage,gray,1);
```

- A partir de la imagen resultante vamos variando el umbral de binarización hasta extraer los contornos.

```

for (int l = 0; l<N; l++)
{

    Imgproc.threshold(gray, gray0, (l+1)*255/N,
    255,Imgproc.THRESH_BINARY);

    List<MatOfPoint> contours=new LinkedList<MatOfPoint>();

    Imgproc.findContours(gray0, contours, new Mat(),
    Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE);

    (...)

}

```

- Finalmente, con los contornos encontrados, se buscan rectángulos aproximando polinomios. También es importante que el rectángulo encontrado esté sobre la vertical de la captura, por lo que se hace necesario detectar el ángulo del polinomio sobre la vertical.

```
public MatOfPoint approxPolyDP(MatOfPoint curve, double epsilon,
boolean closed) {
    MatOfPoint2f tempMat=new MatOfPoint2f();
    Imgproc.approxPolyDP(new MatOfPoint2f(curve.toArray()),
tempMat, epsilon, closed);
    return new MatOfPoint(tempMat.toArray());
}
```

```
public double angle(Point pt1, Point pt2, Point pt0) {
    double dx1 = pt1.x - pt0.x;
    double dy1 = pt1.y - pt0.y;
    double dx2 = pt2.x - pt0.x;
    double dy2 = pt2.y - pt0.y;
    return (dx1*dx2 + dy1*dy2)/Math.sqrt((dx1*dx1 + dy1*dy1)*(dx2*dx2 +
dy2*dy2) + 1e-10);
}
```

6.3.5.4 Implementación de la corrección de ángulos “Deskew”

OpenCV como tal no incluye ninguna función para la detección y corrección de ángulos. Es por ello que se tuvo que implementar la función desde cero partiendo de la base de que el OCR no puede leer caracteres que se encuentren girados o doblados.

Se trata de una detección de líneas horizontales en la imagen, y a partir de su posición se calcula el ángulo de giro. Finalmente se gira la imagen tantos grados como la detección de líneas esté con respecto a la vertical. Aquí se detalla el código fuente:

```

public Mat deskew(Mat imginput) {

    Mat ret = imginput.clone();

    Size size = imginput.size();

    Core.bitwise_not(imginput, ret);

    Point center = new Point(imginput.width()/2, imginput.height()/2);

    Mat lines = new Mat();

    Imgproc.HoughLinesP(ret, lines, 1, Math.PI / 180, 100, size.width/
2.f, 20);

    double angle = 0;

    for (int i= 0; i<lines.height(); i++){

        for (int j= 0; j<lines.width(); j++) {

            angle += Math.atan2(lines.get(i, j)[3] - lines.get(i,
j)[1], lines.get(i, j)[2] - lines.get(i, j)[0]);

        }

    }

    angle /= lines.size().area();

    angle = angle * 180 / Math.PI;

    if (angle >1 || angle <-1){

        Mat rotation = Imgproc.getRotationMatrix2D(center, angle,
1.0);

        Imgproc.warpAffine(imginput, ret, rotation, size);

        return ret;

    }

    return imginput;

}

```

6.4 Implementación del servidor bluetooth en Raspberry Pi.

Como actuador se ha elegido Raspberry Pi y script en Python que nos servirá de servidor Bluetooth donde daremos interpretación a los comandos enviados por la aplicación móvil en forma de señales a través de los puertos GPIO de la Raspberry Pi. Para este proyecto fin de máster, se ha elegido el modelo Raspberry Pi 3B que viene con interfaz Bluetooth incluida. En caso contrario, se debe disponer de un dispositivo externo que haga de interfaz bluetooth para la recepción de los comandos enviados por la aplicación instalada en el Smartphone.

6.4.1 Instalación de *Raspbian* usando NOOBS.

Si acabamos de adquirir una Raspberry Pi, se hace necesaria la instalación de un sistema Operativo que gestione el hardware. Para ello se recomienda usar NOOBS. NOOBS permite instalar de forma sencilla una serie de Sistemas operativos compatibles con Raspberry Pi. Para ello es necesario una tarjeta SD formateada en NTFS.

Descargamos la última versión de la página oficial de Raspberry Pi (<https://www.raspberrypi.org/downloads/noobs/>) y recomendamos descargar la versión NOOBS Lite con el que usaremos una conexión a internet en la Raspberry Pi para descargarnos in situ el sistema operativo *Raspbian*.

Descomprimos los archivos dentro de la Tarjeta microSD que se vaya a usar en la Raspberry Pi.

Una vez copiados los archivos en la tarjeta SD, la insertamos en nuestra Raspberry Pi y la conectamos a la energía y a su vez los periféricos necesarios (pantalla, teclado, ratón...etc.).

Una vez iniciado, seleccionamos *Raspbian* como sistema operativo:

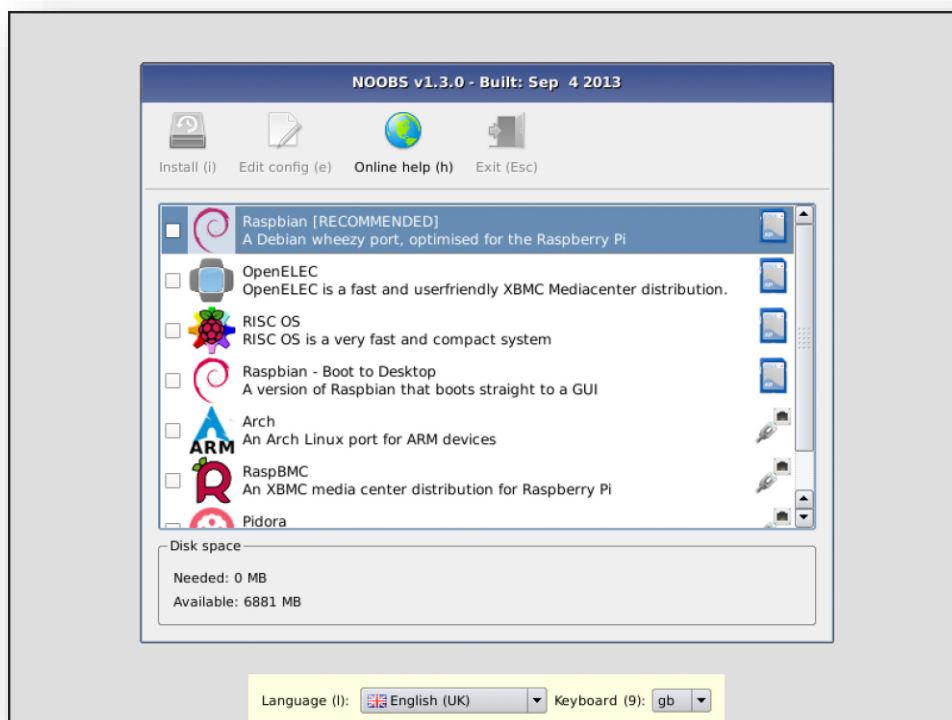


Ilustración 44: Instalación de Raspbian

La instalación es automática una vez descargado, así que tan solo hay que esperar a que se complete el proceso y una vez reiniciada la Raspberry Pi tendremos disponible el entorno de escritorio.

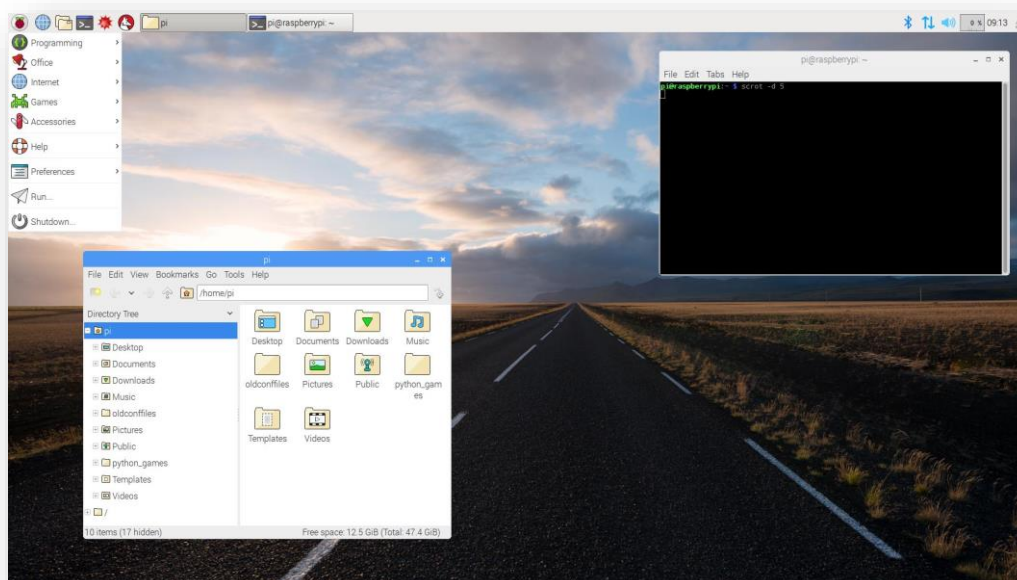


Ilustración 45: Entorno de escritorio Raspbian con Pixel

6.4.2 Preparación del entorno.

Una vez instalado Raspbian, hay que preparar el sistema operativo para operar con la interfaz bluetooth. Para ello hay que descargar los siguientes paquetes:

- Instalar los paquetes que activan el hardware dependiente de bluetooth:

```
:~$ sudo apt-get install pi-bluetooth
```

- Instalar las librerías que manejan el hardware

```
:~$ sudo apt-get install bluetooth bluez
```

- Instalar las librerías de Python necesarias

```
:~$ sudo apt-get install bluez python-bluez
```

- Activar Bluetooth para ser descubierto por un dispositivo externo.

```
:~$ sudo hciconfig hci0 piscan
```

Una vez instalados los paquetes necesarios hay que configurar un fichero para activar la api de compatibilidad (en caso de que se use una de las últimas versiones de Raspbian)

Para ello Editar el siguiente fichero:

```
:~$ gedit "/lib/systemd/system/bluetooth.service"
```

Se debe cambiar la línea :

```
ExecStart=/usr/lib/bluetooth/bluetoothd
```

por:

```
ExecStart=/usr/lib/bluetooth/bluetoothd --compal
```

Una vez configurado el fichero, ya podremos directamente ejecutar el script de Python con permisos de super usuario.

```
:~$ sudo python severRFCOMMBluetooth.py
```

6.4.3 Vistazo al código fuente.

El código es consiste en dos bucles, uno de ellos infinito, que espera eternamente a un mensaje por la interfaz bluetooth.

```
# file: GoKitchen-serverRFCOMMBluetooth.py
# auth: Marc Bayon <mb154@hotmail.com>
# desc: Rfcomm server for GPIO handling.
#

def main():
    while True:

        (...)

        server_sock=BluetoothSocket( RFCOMM )
        server_sock.bind(("",PORT_ANY))
        server_sock.listen(1)

        port = server_sock.getsockname()[1]

        uuid = "94f39d29-7d6d-437d-973b-fba39e49d4ee"

        advertise_service( server_sock, "SampleServer",
                            service_id = uuid,
                            service_classes = [ uuid, SERIAL_PORT_CLASS
                            ],
                            profiles = [ SERIAL_PORT_PROFILE ],
                            )

        print("Waiting for connection on RFCOMM channel %d" % port)

        client_sock, client_info = server_sock.accept()
        print("Accepted connection from ", client_info)

        try:
            while True:
                data = client_sock.recv(1024)
                if len(data) == 0: break
                print("received [%s]" % data)
                if data == "STATUS":
                    print ("Connected")
                    GPIO.output(GPIO_BT_CONNECTED,GPIO.HIGH)
                    server_sock.send("OK")
                    print("Send [OK]")

                if data == "ON":
                    print ("Vitro on")
                    GPIO.output(GPIO_ON_OFF,GPIO.HIGH)
                    POWER_ON = True
                    server_sock.send("ON_OK")
                    print("Send [ON_OK]")
                    buzzer(1)
```

```

        if data == "OFF":
            print ("Vitro OFF")
            clearGPIO(False)
            HOB_POWER = 0
            POWER_ON = False
            server_sock.send("OFF_OK")
            print("Send [OFF_OK]")
            buzzer(2)

        if data == "PWUP":
            print ("Power UP")
            if POWER_ON:
                HOB_POWER = HOB_POWER+1
                if DisplayPower(HOB_POWER):
                    server_sock.send("PWUP_OK")
                    print("Send [PWUP_OK]")
                    buzzer(HOB_POWER)
                else :
                    HOB_POWER = HOB_POWER -1
                    server_sock.send("PWUP_NOK")
                    print("Send [PWUP_NOK]")
            else:
                server_sock.send("PWUP_NOK")
                print("Send [PWUP_NOK]")

        if data == "PDOWN":
            if POWER_ON:
                print ("Power DOWN")
                HOB_POWER = HOB_POWER-1
                if DisplayPower(HOB_POWER):
                    server_sock.send("PDOWN_OK")
                    print("Send [PDOWN_OK]")
                    buzzer(HOB_POWER)
                else :
                    HOB_POWER = HOB_POWER +1
                    server_sock.send("PDOWN_NOK")
                    print("Send [PDOWN_NOK]")
            else:
                server_sock.send("PDOWN_NOK")
                print("Send [PDOWN_NOK]")

    except IOError:
        pass

    print("disconnected")
    clearGPIO(True)

    client_sock.close()
    server_sock.close()
    print("all done")

(...)

if __name__ == "__main__":
    main()

```

6.4.4 Mapa de GPIO Ports usados y esquemático del circuito.

Aquí se representa el mapeo de los GPIO Ports usados para iluminar el display de leds que mostrará la simulación de las señales que deben ser en un futuro conectadas a la placa vitrocerámica real. En esta memoria se ha simulado este comportamiento con Leds y un Buzzer.

Led Comando	GPIO Pin Raspberry Pi
STATUS	GPIO 21
ON	GPIO 20
POT 1	GPIO 23
POT 2	GPIO 24
POT 3	GPIO 25
POT 4	GPIO 12
POT 5	GPIO 16
BUZZER	GPIO 18

Tabla 64: Relación de señales y GPIO Ports.

El montaje del circuito se muestra en la siguiente imagen:

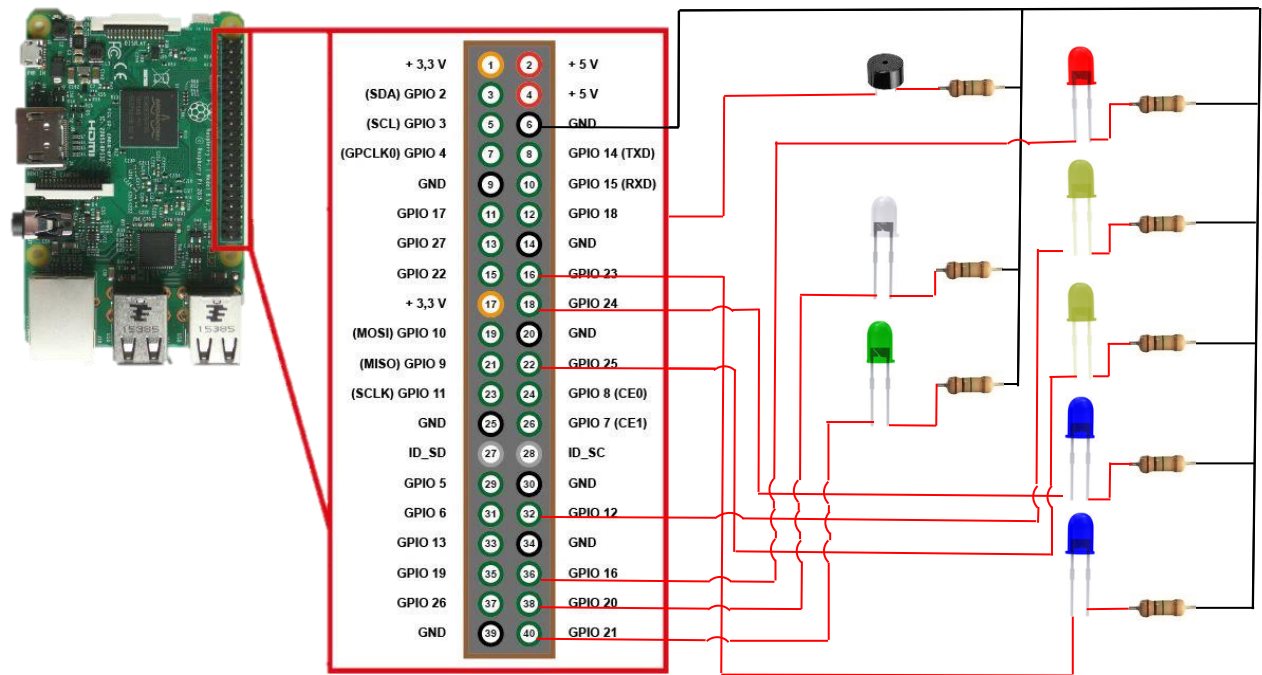


Ilustración 46: Esquemático del circuito de prueba.

CAPÍTULO 7: PRUEBAS DEL SISTEMA

7.1 Introducción

En este capítulo se van a detallar las pruebas del sistema que se han llevado a cabo justo al dar por finalizado la implementación de la aplicación. En este capítulo se va a detallar las pruebas realizadas tanto al subsistema 1 como al subsistema 2 detallados en el capítulo 4.2. El detalle del plan de pruebas y de los dispositivos utilizados se detalla en la siguiente sección:

7.2 Hardware utilizado para el plan de pruebas.

Para el plan de pruebas resultara lo más completo posible se han utilizados dos teléfonos inteligentes, uno de ellos actual para la fecha de redacción de esta tesis adquirido en el año 2016, y otro de ellos con hardware más antiguo datando del año 2010. A continuación, se detallan los datos técnicos de los dispositivos:

- Dispositivo con mayor capacidad, en el plan de pruebas será denominado como “**dispositivo 1**”.

Hardware	Características
Modelo	One Plus II
Lanzamiento	11 de agosto de 2016
Sistema Operativo	Android 6.0.1 basado en OxygenOS
CPU	Octa-core 64-bit 20 nm 1.77 GHz
Chip	Qualcomm Snapdragon 810
GPU	Adreno 430 650 MHz
Almacenamiento	64 GB eMMC v5.0 – 4GB RAM
Cámara trasera	13 MP OmniVision Technologies PureCel-S™ OV13860 1.3µm-pixel optical sensor, f/2.0 aperture, 2160p@30fps, 1080p@60fps, 720p@120fps, estabilizador óptico de imagen, laser autofocus, dual LED flash
Bluetooth	Versión 4.1

- Dispositivo con menor capacidad, en el plan de pruebas será denominado como “**dispositivo 2**”.

Hardware	Características
Modelo	Google Nexus S
Lanzamiento	16 de diciembre de 2010
Sistema Operativo	Android 4.1.2 Jelly Bean
CPU	Single-core, 45nm 1 GHz
Chip	Samsung Exynos 3110 (ARM Cortex A8)
GPU	-
Almacenamiento	16 GB / 1GB RAM
Cámara trasera	5 Megapixel sin tecnología adicional
Bluetooth	Versión 2.1

7.3 Plan de pruebas a realizar

El plan de pruebas consiste en probar la lectura con una cámara de alta definición y con otra de baja resolución. Por otro lado, medir el tiempo de procesamiento y la conectividad Bluetooth en términos de distancia máxima y probar el dispositivo en caso de encontrarnos con errores de protocolo o de mensajes no recibidos.

7.3.1 Pruebas de lectura OCR

El plan de pruebas de lectura OCR se basa en los siguientes hitos:

- Test de luminosidad

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Lectura con luz ambiente, luz solar	✓	✓	Dificultad en ambos dispositivos para Detectar la ROI si la silueta del teléfono se ve reflejada en la vitrocerámica. D1: 83% Éxito D2: 65 % Éxito
Lectura con baja luminosidad y luz artificial	✓	~	D2: dificultad en definir la ROI por la baja diferencia de color entre la ROI y el fondo.
Lectura con alta luminosidad, luz artificial	✓	✓	No se han detectado anomalías. El reflejo del teléfono puede ser un problema dependiendo de la orientación de la fuente de luz

Ilustración 47: Resultados del Test de luminosidad

- Test de distancia de lectura

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Lectura a 10 cm	✓~	✓~	Distancia mínima D1: 7 cm D2 6 cm A menos de 5 cm la ROI se sale fuera del alcance del enfoque de la cámara. Baja fiabilidad en ambos dispositivos.
Lectura a 15 cm	✓	✓	Fiabilidad: D1: 80% D2: 70% Es la distancia con mejores resultados
Lectura a 20 cm	~	~	Apenas se consiguen resultados positivos. D1: 22% fiabilidad D2: 9% fiabilidad
Lectura a 30 cm	X	X	La máxima lectura con escasa fiabilidad fue: D1: 25 cm D2: 21 cm

Tabla 65: Resultados del test de distancia de lectura

- Test de lectura con posición distinta a la horizontal

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Lectura normal con 0° de inclinación sobre la vertical	✓	✓	
Lectura normal con 45° de inclinación sobre la vertical	X	X	Detección de ROI buena en ambos dispositivos. La lectura no ha sido positiva debido a que el filtrado no ha sido exitoso para líneas verticales y horizontales.
Lectura normal con 90° de inclinación sobre la vertical	X	X	Detección de ROI buena en ambos dispositivos. La lectura no ha sido positiva debido a que el filtrado no ha sido exitoso para líneas verticales y horizontales.
Lectura normal con 135° de inclinación sobre la vertical	X	X	No detecta ROI ni lectura
Lectura normal con 180° de inclinación sobre la vertical	X	X	No detecta ROI ni lectura
Lectura invertida	X	X	No detecta ROI ni lectura

Tabla 66: Resultados del test de lectura en distintas posiciones.

- Test con distintas tipografías 7 Segmentos

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Test con Tipografía 1	✓	✓	
Test con tipografía 2	✓	X	El entrenamiento solo ha sido realizado sobre un tipo de tipografía. Debería hacerse para otros tipos
Test con tipografía 3	X	X	El entrenamiento solo ha sido realizado sobre un tipo de tipografía. Debería hacerse para otros tipos
Test con tipografía 4	X	X	El entrenamiento solo ha sido realizado sobre un tipo de tipografía. Debería hacerse para otros tipos
Test Con tipografía Manuscrita	X	X	El entrenamiento solo ha sido realizado sobre un tipo de tipografía. Debería hacerse para otros tipos

Tabla 67: Resultado del test con distintas tipografías.

- Test sobre la detección del área de reconocimiento (ROI)

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Perímetro con área definida correctamente	✓	✓	
Perímetro con área definida parcialmente	~	~	Conseguido con un 50% de fiabilidad en ambos dispositivos
Perímetro con área definida con un color distinto al blanco	✓	✓	
Test de lectura de área con máximo tamaño	✓	✓	D1: 9cm D2: 6cm
Test de lectura con área de mínimo tamaño	✓	✓	D1: 25 cm D2: 21 cm

Tabla 68: Resultado del test de detección de la ROI.

7.3.2 Test de conexión Bluetooth

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Envío de mensaje normal a distancia de 5-10 m	✓	X	D2: No es posible la conexión debido a que no es compatible con Bluetooth 3.0
Envío de mensaje a distancia 50-60 m	✓	X	D2: No es posible la conexión debido a que no es compatible con Bluetooth 3.0
Envío de mensaje corrupto	✓	X	D2: No es posible la conexión debido a que no es compatible con Bluetooth 3.0
Envío de mensaje incompleto	✓	X	D2: No es posible la conexión debido a que no es compatible con Bluetooth 3.0

Tabla 69: Resultado del test de conexión bluetooth.

7.3.3 Test sobre el modo accesibilidad con Android Gesture.

Descripción del Test	Resultado dispositivo 1	Resultado dispositivo 2	Observaciones
Pruebas de todos los gestos con similitud al patrón	✓	✓	
Prueba de todos los gestos con una similitud algo diferente al patrón	✓	~	D2: Fue diseñado anterior a la aparición de Android Gesture
Prueba de gestos totalmente diferentes al patrón	✓	~	D2: Fue diseñado anterior a la aparición de Android Gesture

Tabla 70: Resultados del test de Android Gesture.

7.3.4 Prueba de esfuerzo en Raspsberry Pi

Descripción del Test	Resultado Raspberry Pi	Observaciones
Prueba de conexión y desconexión x 1000 ciclos	✓	Ok
Prueba de reinicio e inicio del servicio x 100 ciclos	✓	Ok
Prueba de recepción continuada de mensajes durante 24 horas.	✓	Ok

Tabla 71: Resultados del test de esfuerzo en Raspberry Pi.

ANEXO1: MANUAL DE USUARIO

A.1 Introducción

En este anexo se explica el manual de usuario de la aplicación. Esta aplicación puede ser instalada en cualquier dispositivo Android y su uso es intuitivo para cualquier persona que esté acostumbrada al manejo de teléfonos inteligentes.

En caso de querer consultar detalles técnicos o ver el código fuente del aplicativo, se recomienda consultar las secciones 4.2 y 6.3 para entrar en más detalles.

Para el funcionamiento completo de la aplicación es necesario de disponer la vitrocerámica con conexión bluetooth. En su lugar el prototipo utilizado para la exposición en su lugar ha sido una raspberry Pi 3B con las señales en forma de Led, para más información consultar el capítulo 6.4.4. de la memoria desarrollada para este trabajo fin de máster.

La aplicación se encuentra disponible en 3 idiomas, español, alemán e inglés.

A.2 Requisitos del teléfono inteligente

Para hacer uso de la aplicación es necesario un dispositivo con las siguientes características:

Software	
Sistema Operativo	Android
Versión mínima	Ice Cream Sandwich v4.0.1
Biblioteca Gráfica	Opencv APP
Tienda de software	Google play
Hardware	
Cámara	Captura de imágenes con una resolución superior a 640x355
Bluetooth	Version 2.1 o superior
Almacenamiento	Entre 15 y 17 MB de almacenamiento interno

Tabla 72: Requisitos de software y hardware.

A.3 Instalación y puesta en marcha

Para instalar la aplicación en el teléfono inteligente basta con acudir a la aplicación Google Play y buscar la aplicación “Go Kitchen”.

Para arrancar el servicio de bluetooth, basta con proporcionar energía a la vitrocerámica con conexión bluetooth o al prototipo Raspberry pi y esperar unos minutos a que se establezca la conexión.

A.4 Modo de empleo y funcionamiento

Antes de arrancar la aplicación es recomendable emparejar el dispositivo con la placa vitrocerámica y establecer el enlace bluetooth para la posterior comunicación.

La aplicación se encuentra localizada tras el icono siguiente:

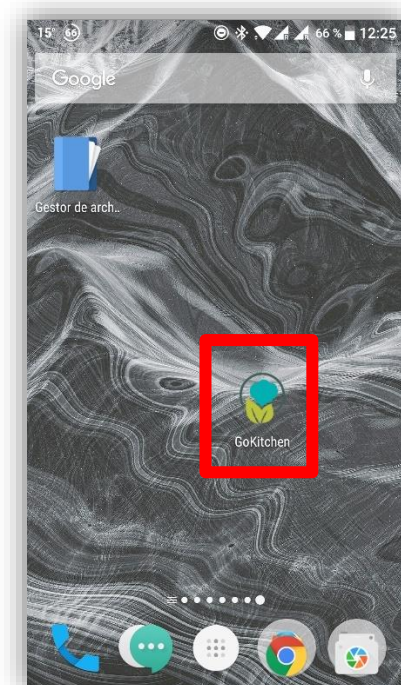


Ilustración 48: Lanzador de la aplicación en Android

Una vez iniciada nos aparecerá la pantalla de *splash*. Esta pantalla se muestra unos segundos y acto seguido debe mostrarse la pantalla principal de la aplicación.

A.5 Modo con botones

Permite el manejo de la aplicación a personas videntes. A continuación, se explica el manejo de los botones:

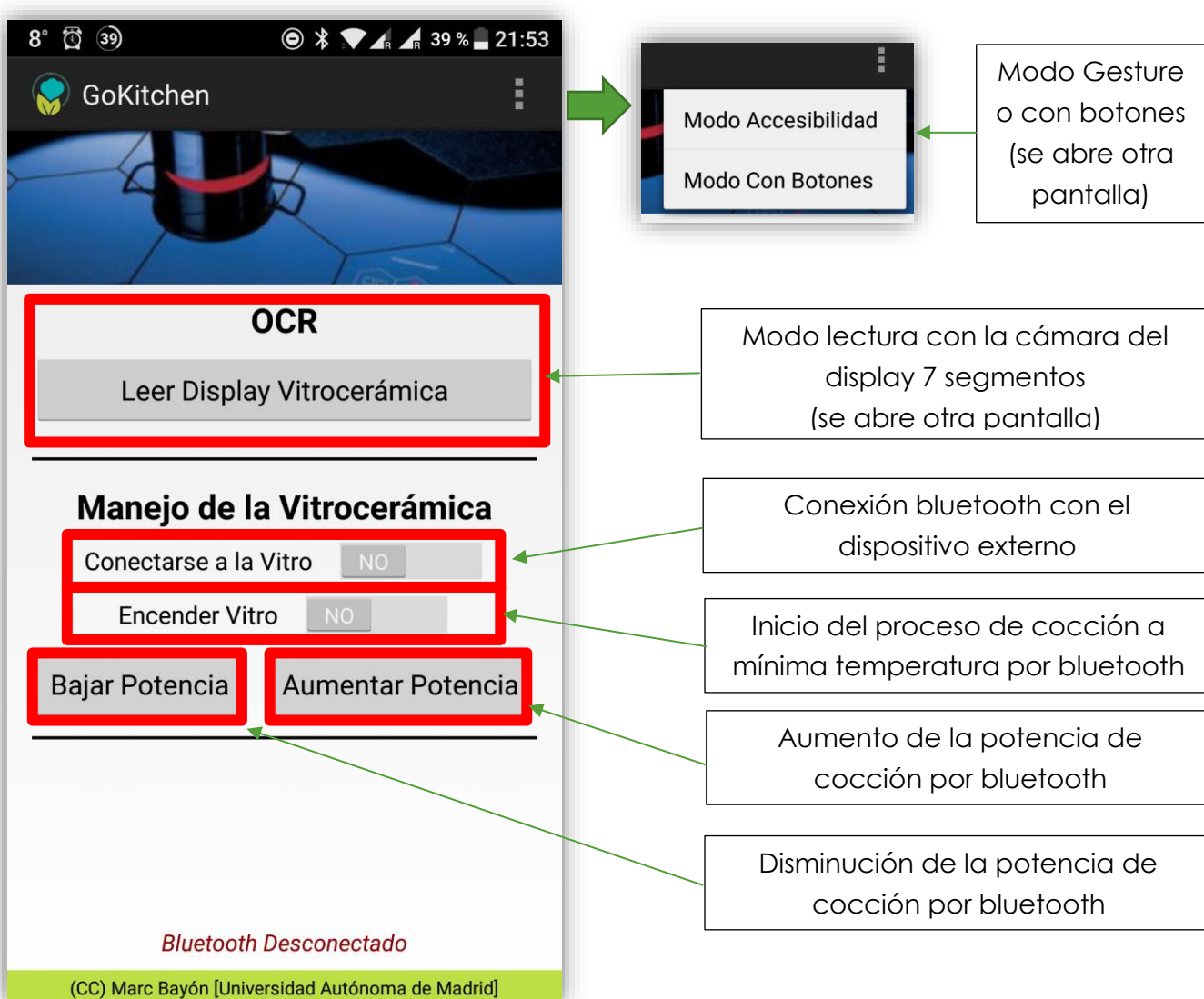


Ilustración 49: Detalle de la pantalla principal de la aplicación.

A.6 Modo accesibilidad (Gesture)

El modo accesibilidad o Gesture permite realizar las mismas acciones que el modo con botones, solo que con gestos. Aquí se muestra la pantalla principal:

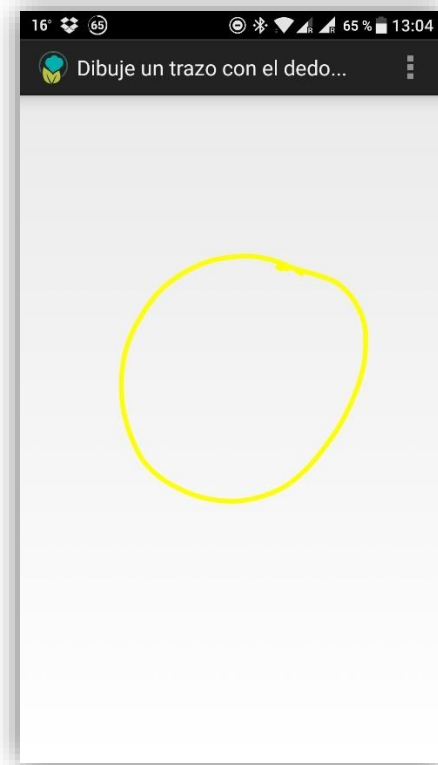


Ilustración 50: Detalle del modo gesture.

Para el modo gesture existen los siguientes comandos disponibles:



Modo OC, para lectura del display 7 segmentos (se abre otra pantalla)



Conectarse a la vitrocerámica



Desconectarse de la vitrocerámica



Empezar la cocción



Finalizar la cocción



Aumentar potencia de cocción



Disminuir potencia de cocción

En caso de finalizar la conexión con la vitrocerámica, esta se desconectará completamente por seguridad.

A.7 Modo OCR

En este modo se procederá a la lectura del display 7 segmentos. Para proceder a la lectura del número tan solo hay que acercar la cámara principal del teléfono al display 7 segmentos de la cocina vitrocerámica. Cuando aparezca un recuadro verde quiere decir que la aplicación ha encontrado una posible zona de lectura, con lo que se debe mantener el teléfono sobre esa posición durante unos segundos. Acto seguido, la lectura aparecerá en la parte posterior de la pantalla y también se anunciará por el altavoz del teléfono (verificar antes que el volumen del altavoz es adecuado para la escucha)

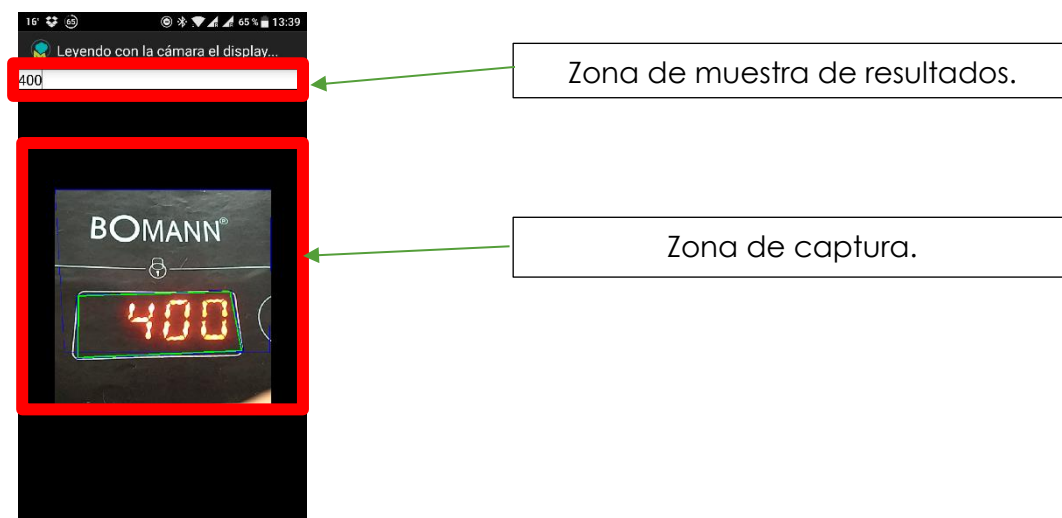


Ilustración 51: Detalle del modo OCR

Apéndice 1: Referencias

- [1] «California license plate ANPR,» [En línea]. Available: https://commons.wikimedia.org/wiki/File:California_license_plate_ANPR.png.
- [2] Sentino, «Sentino Cook System,» [En línea]. Available: <http://www.tuvie.com/sentino-cook-top-for-visually-impaired-people/>.
- [3] Wikipedia, «COCOMO,» [En línea]. Available: <https://es.wikipedia.org/wiki/COCOMO>.
- [4] OpenCV, «OpenCV Documentation,» 2017. [En línea]. Available: <http://docs.opencv.org/2.4/index.html>.
- [5] OpenCV, «OpenCV-Structural Analysis and Shape Descriptors,» OpenCV, 2015. [En línea]. Available: http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html.
- [6] J. Morris, «Tesseract-Improving the quality of the output,» 30 07 2016. [En línea]. Available: <https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality>. [Último acceso: 1 08 2016].
- [7] N. Otsu, «A Threshold Selection Method from Gray-Level Histograms,» *IEEE*, Vols. %1 de %20018-9472 , nº 2168-2909 , p. 66, 1979.
- [8] OpenCv, «Smoothing Images,» [En línea]. Available: http://docs.opencv.org/3.2.0/d4/d13/tutorial_py_filtering.html.
- [9] F. Abecassis, «OpenCV - Detect skew angle,» 18 09 2011. [En línea]. Available: <http://felix.abecassis.me/2011/09/opencv-detect-skew-angle/>.
- [10] B. S. Proprietary, «Bluetooth Core Specification,» 06 12 2016. [En línea]. Available: <https://www.bluetooth.com/specifications/adopted-specifications>. [Último acceso: 01 02 2017].
- [11] I. e. I. Wikipedia, «Binary Protocol,» Wikipedia., 15 Enero 2017. [En línea]. Available: https://en.wikipedia.org/wiki/Binary_protocol. [Último acceso: 5 Mayo 2017].
- [12] Leptonica , «Leptonica,» Google, 2017 Enero 4. [En línea]. Available: <http://www.leptonica.com/>.
- [13] R. Theis, «Tess-two an Android wrapper for Tesseract,» [En línea]. Available: <https://github.com/rmtheis/tess-two>.
- [14] S. Lian, Java Native Interface: Programmer's Guide and Specification, Addison-Wesley, 1999.
- [15] W. Lim, «A Beginner's Guide to Wireframing,» evantotuts, 18 Junio 2012. [En línea]. Available: <https://webdesign.tutsplus.com/articles/a-beginners-guide-to-wireframing--webdesign-7399>. [Último acceso: 09 05 2017].
- [16] A. Developers, «What is Android?,» Android Inc., 2009 julio 2009. [En línea]. Available: <https://developer.android.com/about/index.html>. [Último acceso: 15 febrero 2017].
- [17] A. Inc., «Logotipo de Android».
- [18] Gartner, «Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016,» Gartner, Inc. and/or its Affiliates, 1 agosto 2016. [En línea]. Available: <http://www.gartner.com/newsroom/id/3415117>. [Último acceso: 15 febrero 2017].

- [19] Tech-thoughts, «Global Smartphone Market Share Trends: Android, iPhone Lead, Windows Phone Struggles,» Tech-Thoughts, 17 julio 2012. [En línea]. Available: <http://www.tech-thoughts.net/2012/07/global-smartphone-market-share-trends.html#.WKSQX-52PE>. [Último acceso: 15 febrero 2017].
- [20] R. Pi, «Raspberry Pi Foundation,» Raspberry Pi Foundation, 1 Enero 2017. [En línea]. Available: <https://www.raspberrypi.org>. [Último acceso: 15 febrero 2017].
- [21] R. Pi, «Raspbian,» Raspberry Pi Foundation, 11 enero 2017. [En línea]. Available: <https://www.raspberrypi.org/downloads/raspbian/>. [Último acceso: 15 febrero 2017].
- [22] LXDE, «Lightweight X11 Desktop Environment,» LXDE Contributors, 25 enero 2017. [En línea]. Available: <http://lxde.org/>. [Último acceso: 15 febrero 2017].
- [23] Microsoft, «Xamarin Inc,» Xamarin Inc, 01 Enero 2017. [En línea]. Available: https://www.xamarin.com/download-it?_bt=101035044668&_bk=xamarin&_bm=e&gclid=CjwKEAiAxKrFBRDm25f60OegtwwSJABgEC-ZZ057le1e4EtfF5FJaeW4YPzpEnxdK_FmUi0pGLb-KRoCeBXw_wcB. [Último acceso: 20 Febrero 2017].
- [24] T. S. T. C. S. Company, «TIOBE Software: The Coding Standards Company,» [En línea]. Available: <http://www.tiobe.com/>. [Último acceso: 20 febrero 2017].
- [25] G. Developers, «Android NDK,» Google, [En línea]. Available: <https://developer.android.com/ndk/index.html?hl=es>.
- [26] G. Developers, «Implementación de Bluetooth en Android,» Google, [En línea]. Available: <https://developer.android.com/guide/topics/connectivity/bluetooth.html?hl=es-419>.
- [27] G. Builder, «Gesture builder Aplicación,» Manan Gandhi, 11 Marzo 2015. [En línea]. Available: https://play.google.com/store/apps/details?id=pack.GestureApp&hl=es_419. [Último acceso: 2017 Mayo 04].
- [28] O. Image Pyramids, «Image Pyramids,» OpenCV, 15 01 2017. [En línea]. Available: http://docs.opencv.org/trunk/dc/dff/tutorial_py_pyramids.html. [Último acceso: 06 05 2017].
- [29] Confederación de Autismo en España, «Confederación de Autismo en España,» [En línea]. Available: <http://www.autismo.org.es/actualidad/articulo/autismo-espana-trabaja-con-autismo-europa-en-la-directiva-europea-de>.
- [30] R. M. Roselló, «Desde mi experiencia,» 2 Febrero 2015. [En línea]. Available: <http://rafamatas.blogspot.co.at/2015/02/ceguera.html>.
- [31] Fagor, «Blog Fagor, Cómo funciona la inducción,» [En línea]. Available: <http://blog.fagor.com/cocina/electrodomesticos/como-funciona-la-induccion/>.
- [32] N. Imsand, «Cooking Without Looking,» [En línea]. Available: <https://www.behance.net/gallery/1009323/Cooking-Without-Looking>.
- [33] J. Garvey, «Touch&Turn: clever cooking solution for the blind,» 30 Noviembre 2009. [En línea]. Available: <http://www.gizmag.com/touchturn-cooking-solution-blind/13482/>.
- [34] G. A. P. Moreno, «The ancient home,» 11 Noviembre 2015. [En línea]. Available: <http://theancienthomesa.blogspot.co.at/2015/11/introduccion-en-este-se-ha-creado-con.html>.

- [35] T. OCR, «Ashton Campion - Utilidad de Google Translator.,» Ashton Campion School of English, 4 febrero 2015. [En línea]. Available: <http://ashtoncampion.com/la-utilidad-de-google-translate/>. [Último acceso: 17 febrero 2017].
- [36] S. Mycosystems, «Java (programming language),» Wikipedia, 20 febrero 2017. [En línea]. Available: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). [Último acceso: 20 febrero 2017].
- [37] A. rewriters, «Bug about missing dependencies in NDK 14++», open source project, 28 Septiembre 2016. [En línea]. Available: <https://android-review.googlesource.com/#/c/280431/>. [Último acceso: 15 Marzo 2017].

